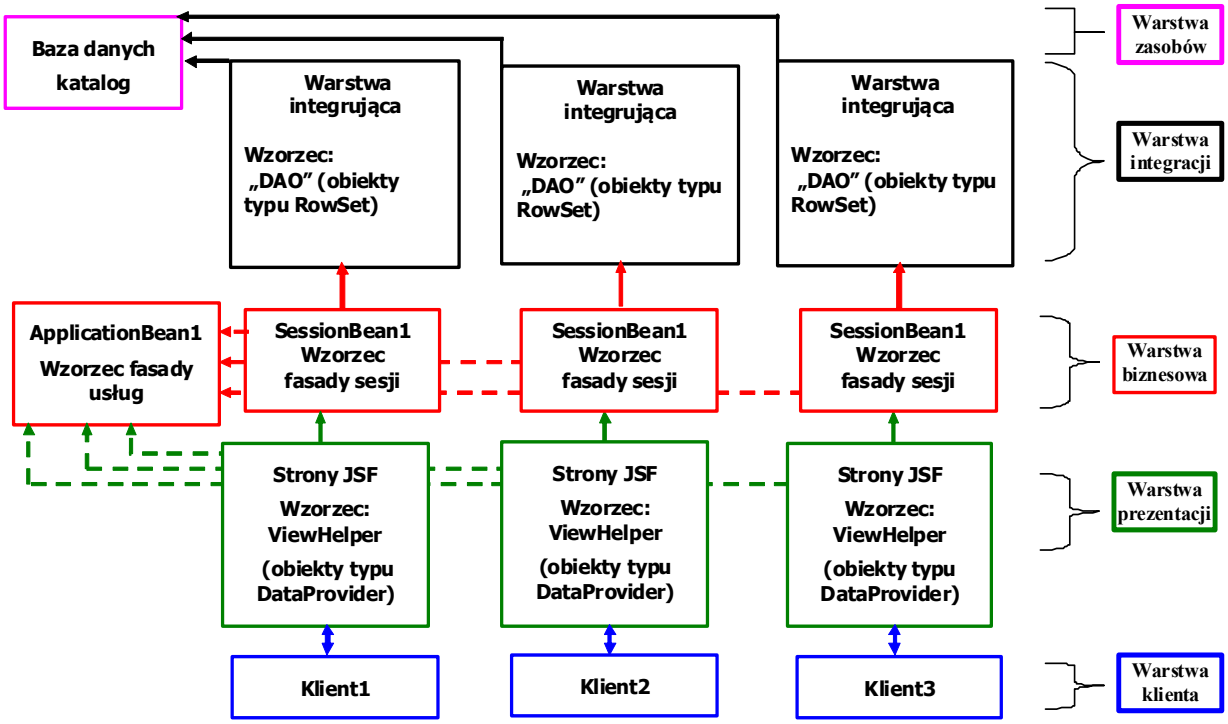
Laboratorium9_10.

Architektura aplikacji pięciowarstwowej (linie przerywane oznaczają powiązania nie wykorzystane w aplikacji). Logika biznesowa jest realizowana za pomocą operacji na bazie danych z wykorzystaniem wzorca DAO.



Wstęp

- Usuń tabele z bazy danych utworzone w aplikacjach z laboratorium5_6 lub laboratorium 7_8.
- Dokonaj zmiany w generowaniu kluczy obcych wg slajdów 1 i 2 (@GeneratedValue(strategy=GenerationType.IDENTITY)). Uruchom aplikacje typu WebApplication z lab.5_6 lub 7_8 (przed uruchomieniem aplikacji typu WebApplication należy skompilować projekt Java Application) i wstaw kilka danych dotyczących tytułów i egzemplarzy w aplikacji. Slajdy 3 i 4 pokazują przykładową nową zawartość bazy danych. Slajd 5 pokazuje właściwości kluczy głównych w obu tabelach.

Przykład 2 - @GeneratedValue(strategy=GenerationType.IDENTITY)

TTytul_ksiazki.java * | TEgzemplarz.java * | persistence.xml *...

```
1  package wypozyczalnia1app;
2
3  import java.io.Serializable;
4  import java.util.ArrayList;
5  import java.util.Collection;
6  import javax.persistence.Entity;
7  import javax.persistence.GeneratedValue;
8  import javax.persistence.GenerationType;
9  import javax.persistence.Id;
10 import javax.persistence.OneToMany;
11 import javax.persistence.Transient;
12
13 @Entity
14 public class TTytul_ksiazki implements Serializable {
15     private static final long serialVersionUID = 1L;
16     @Id
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Long id;
19     private String wydawnictwo;
20     private String ISBN;
21     private String tytul;
22     private String autor;
23     @OneToMany(mappedBy = "mTytul_ksiazki")
24     private Collection<TEgzemplarz> Ksiazka;
25     @Transient
26     private ArrayList<TEgzemplarz> mKsiazka =
27             new java.util.ArrayList<TEgzemplarz>();
```

12:1 INS

Slajd 1.

Wypozyczalnia1app - NetBeans IDE 6.0.1

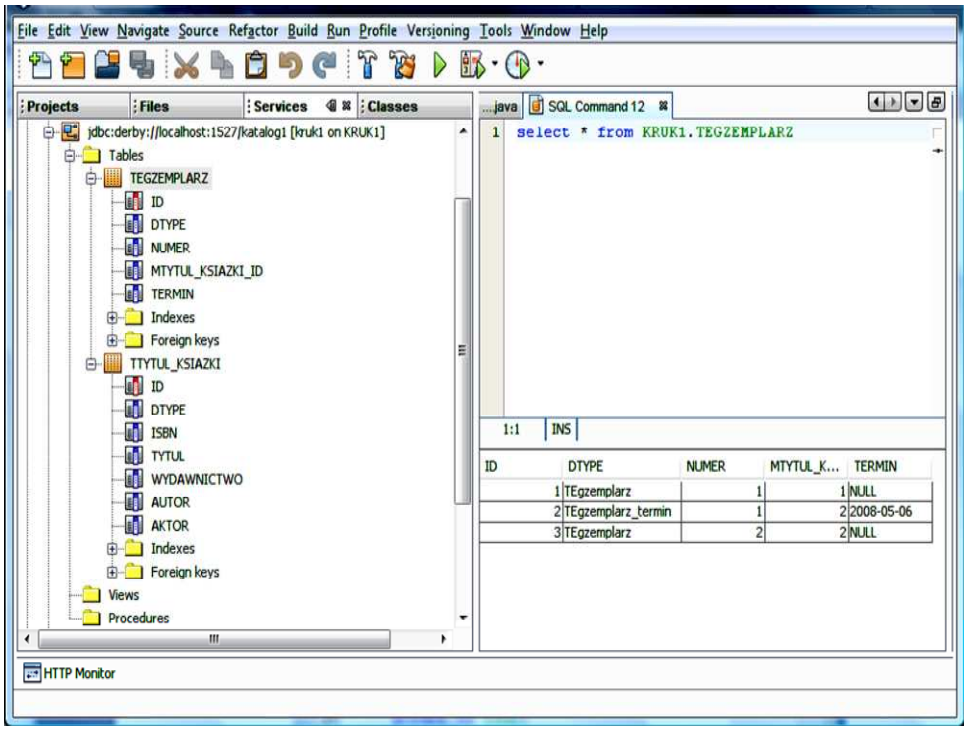
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

...java * | TEgzemplarz.java * | persistence.xml * | SQL Command 9...

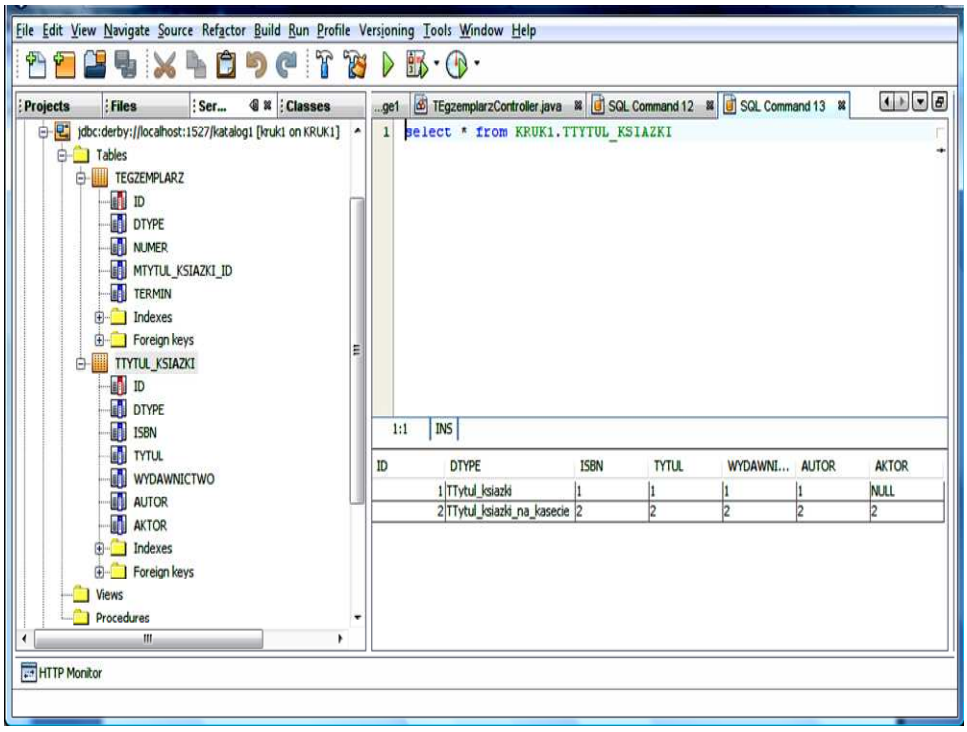
```
1  package wypozyczalnia1app;
2
3  import java.io.Serializable;
4  import java.util.Date;
5  import javax.persistence.Entity;
6  import javax.persistence.GeneratedValue;
7  import javax.persistence.GenerationType;
8  import javax.persistence.Id;
9  import javax.persistence.ManyToOne;
10
11 @Entity
12 public class TEgzemplarz implements Serializable {
13     private static final long serialVersionUID = 1L;
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Long id;
17     private int numer;
18     @ManyToOne
19     private TTytul_ksiazki mTytul_ksiazki;
```

15:55 INS

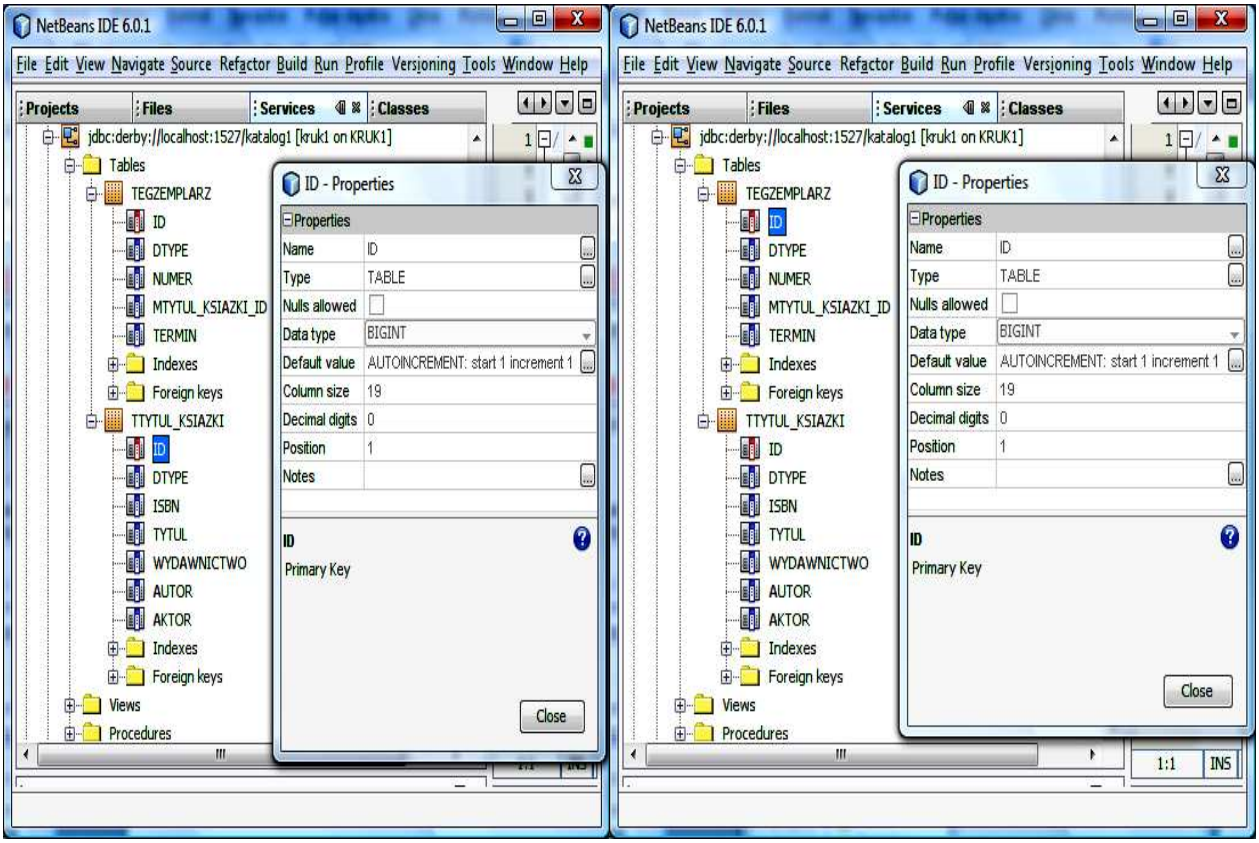
Slajd 2.



Slajd 3.



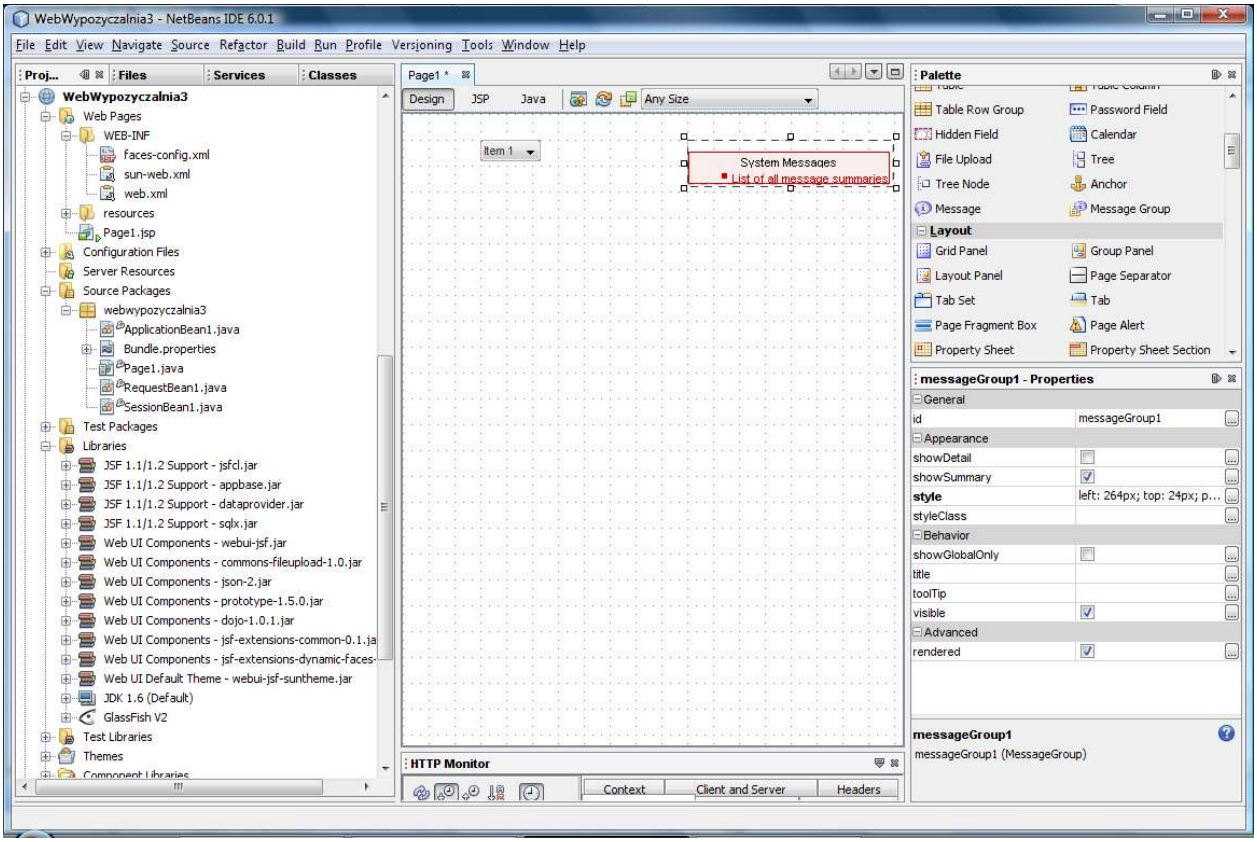
Slajd 4.


NetBeans IDE 6.0.1
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Projects
Files
Services
Classes
jdbc:derby://localhost:1527/katalog1 [kruk1 on KRUK1]
Tables
TEGZEMPLARZ
ID
DTYPE
NUMER
MTYTUL_KSIAZKI_ID
TERMIN
Indexes
Foreign keys
TTYTUL_KSIAZKI
ID
DTYPE
ISBN
TYTUL
WYDAWNICTWO
AUTOR
AKTOR
Indexes
Foreign keys
Views
Procedures
ID - Properties
Properties
Name ID
Type TABLE
Nulls allowed
Data type BIGINT
Default value AUTOINCREMENT: start 1 increment 1
Column size 19
Decimal digits 0
Position 1
Notes
ID
Primary Key
Close
1:1
INS

Slajd 5.

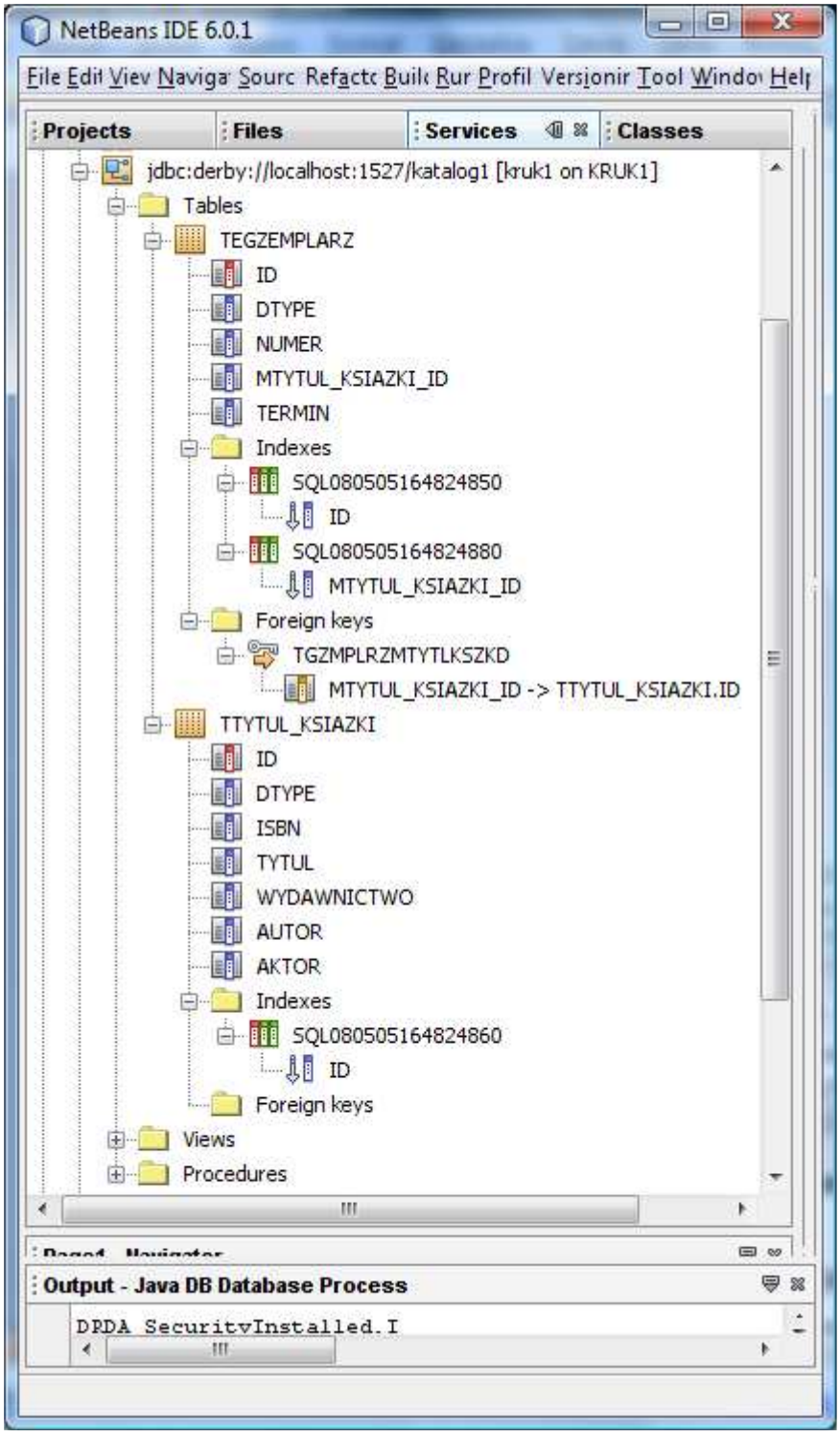
Dwie tabele powiązane relacją jeden do wiele – Baza danych katalog1: tabela TTytul_ksiazki w relacji jeden do wiele do tabeli TEgzemplarz

1. Wybierz opcję File/New Project. Wybierz kategorię projektu Web, a rodzaj projektu Web Application
2. Kliknij na Next
3. Podaj nazwę projektu (Project name), wybierz katalog (Project Location)- np. WebWypozyczalnia3
4. Wybierz serwer aplikacji (Server – GlassFish V2) oraz wersję Javy Enterprise (Java EE Version – Java EE 5).
5. Naciśnij Next – w nowym formularzu należy wybrać Framework Visual Web JavaServer Faces
6. Okno projektu (Projects) zawiera układ plików typu BluePrints. Plik Page1.jsp jest stroną startową napisaną w języku JSP.
7. Okno Files zawiera układ fizyczny plików. Plik Page1.jsp oraz Page1.java stanowią całość- plik Page1.java obsługuje główną stronę jsp.
8. Zaprojektuj stronę Page1 korzystając z Palette Basic
   7.1. Przeciągnij komponent Drop Down List. W oknach Properties id komponentu jest standardowe: dropDown1. Identyfikator id można zmieniać.
   7.2. Przeciągnij komponent Message Group, który będzie wyświetlał komunikaty metod info(*String*), error(*String*), warn(*String*), lub fatal(*String*). Domyślnie, wyświetlane są komunikaty typu błędy wykonania, błędy walidacji oraz błędy konwersji.

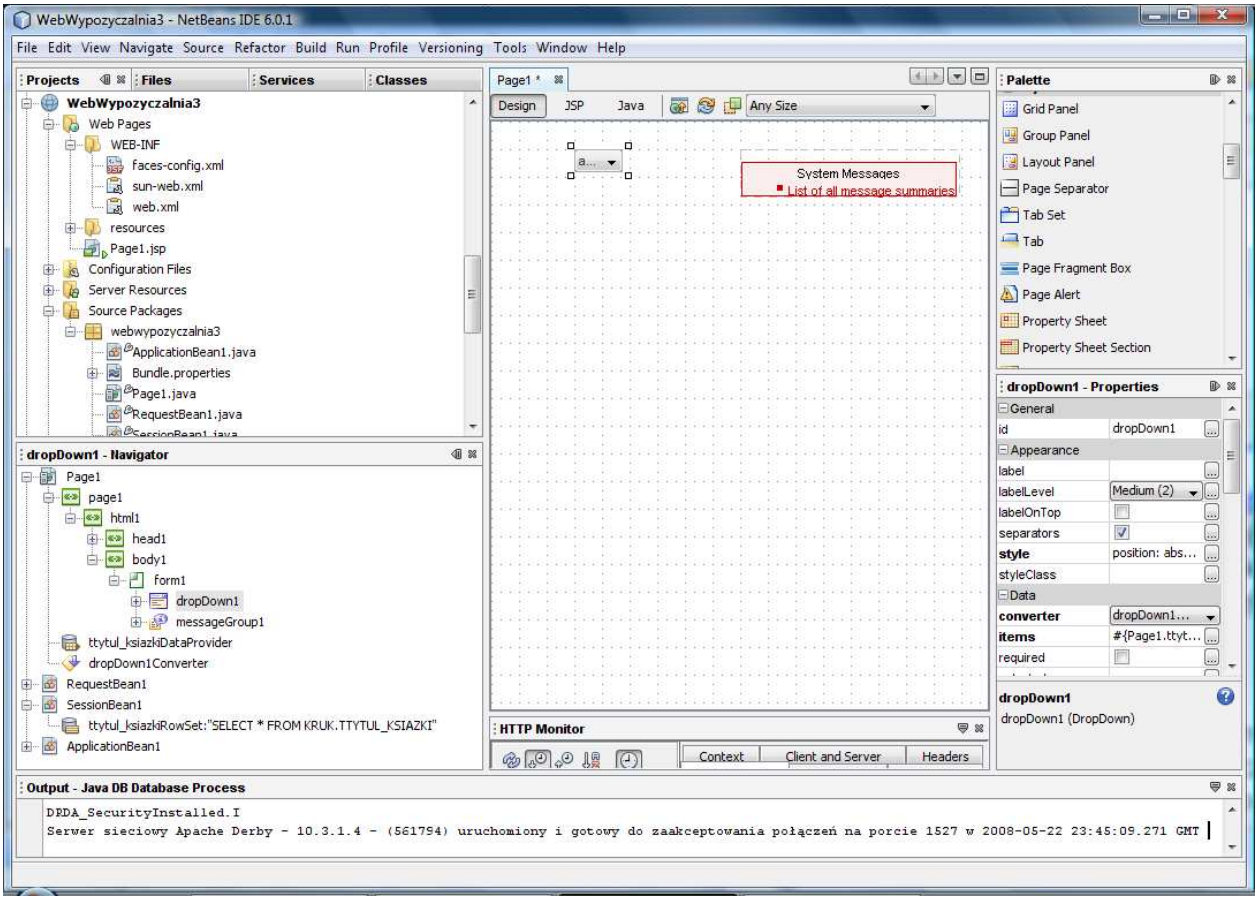


   7.3. Połącz komponent dropDown1 z bazą danych katalog1. W tym celu należy kliknąć na zakładkę Services, następnie rozwinąć opcję

Databases (kliknąć na przycisk +) i prawym klawiszem myszy kliknąć na bazę danych jdbc:derby://localhost:1527/katalog1[kruk1 on Kruk1]. Na wyskakującym menu kliknąć na opcję Connect. .

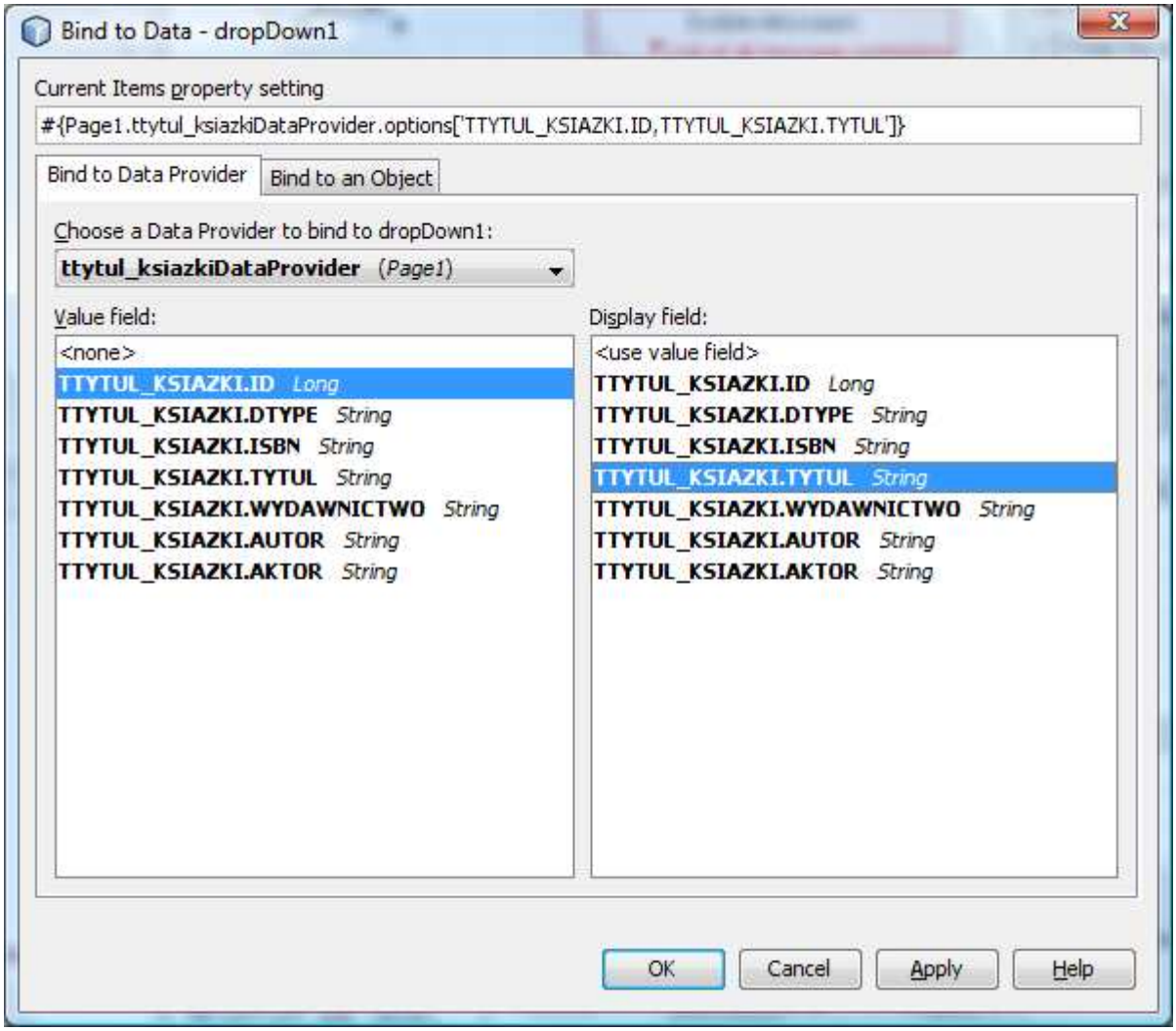


7.4. Przeciągnij tabelę TTytul_ksiazki z zakładki Services na komponent dropDown1. Na komponencie pokazały się elementy listy typu abc, co oznacza połączenie z łańcuchami typu varchar kolumn wybranej tabeli TTytul_ksiazki.

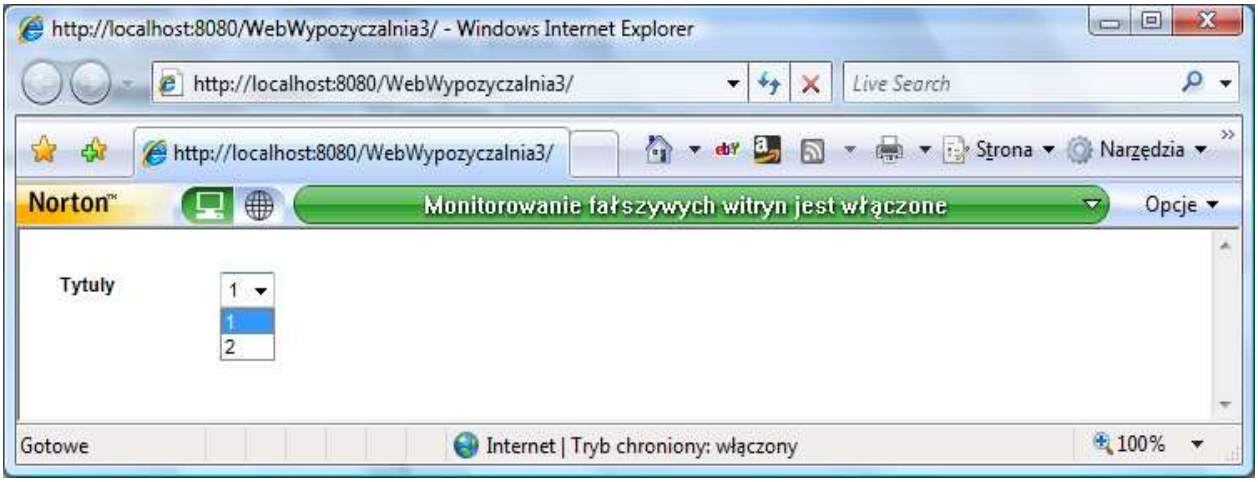


Pojawił się komponent ttytul_ksiazkiDataProvider odpowiedzialny za połączenia z warstwa biznesową (obiekty EJB, tablice Arrays) oraz właściwości ttytul_ksiazkiRowSet w obiekcie SessionBean1, odpowiedzialne za połączenie i obsługę zapytań do tabel w bazie danych (wykonanie zapytania oraz zarządzanie jego wynikiem).

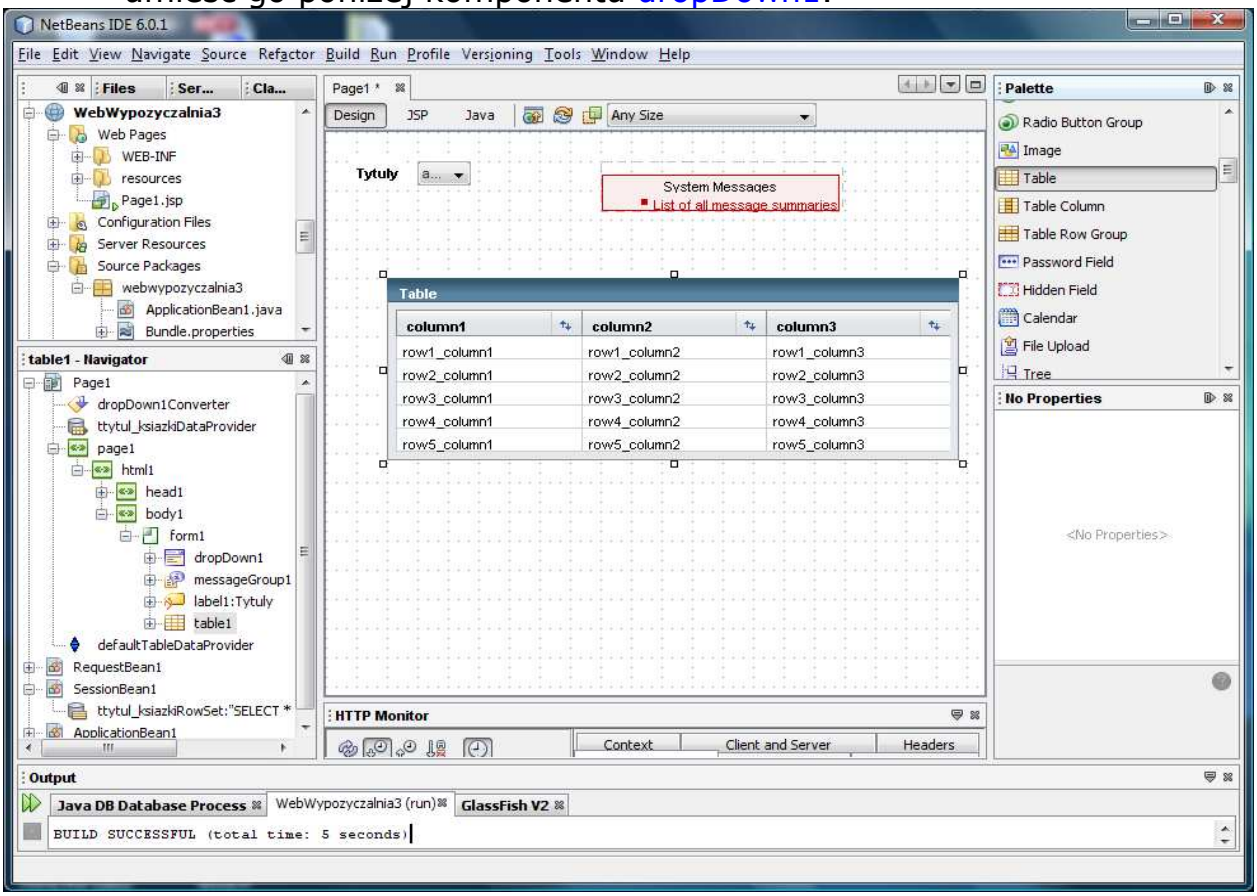
7.5. Prawym klawiszem myszy kliknąć na komponencie dropDown1 i z wyskakującego menu wybrać Bind To Data. W wywołanym formularzu wybrać zakładki Value oraz Field i zaznaczyć TTYTUL_KSIAZKI.ID jako domyślną wartość w programie oraz w zakładce Display wybrać TTYTUL_KSIAZKI.TYTUL jako kolumnę do wyświetlania na pozycjach komponentu dropDown1. Umieść kolejny element label1 typu Label z lewej strony komponentu dropDown1 i wpisz na nim nazwę Tytuly (właściwość text w okienku Properties).



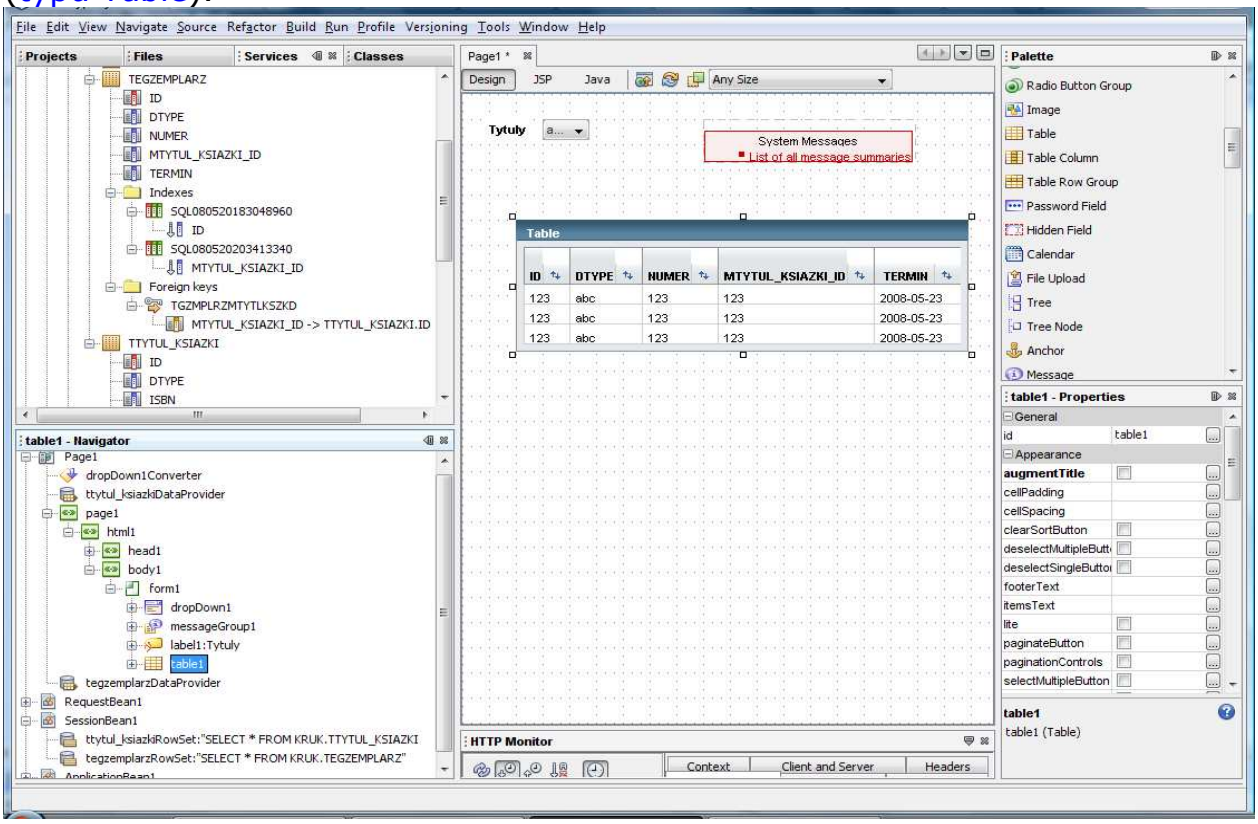
7.6. Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie Project na nazwę projektu, w ukazanym oknie uruchom kolejno Build Project, Deploy Project, Run Project lub tylko Run) i uruchom poszczególne funkcje aplikacji.)



7.7. Przeciągnij komponent Table na stronę Page1 w trybie Design i umieść go poniżej komponentu dropDown1.

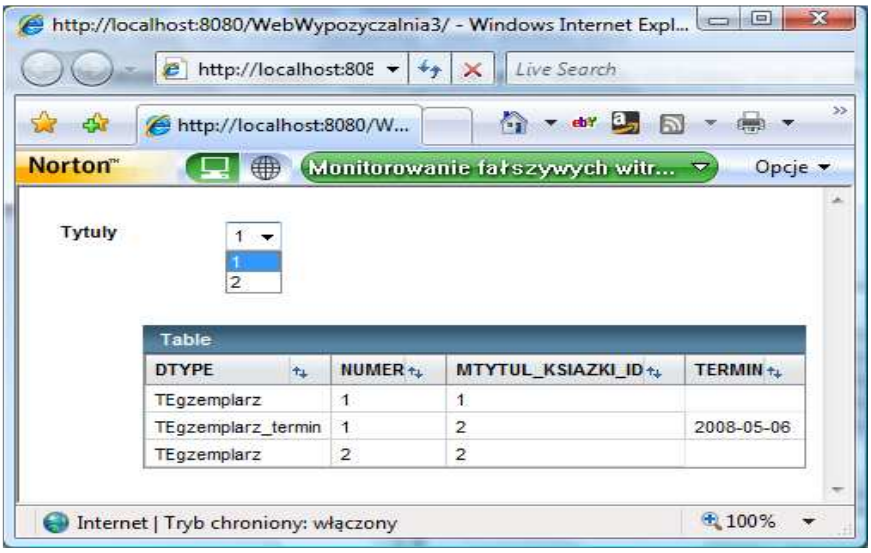


W oknie Services przeciągnij tabelę TEGZEMPLARZ na komponent table1 (typu Table).



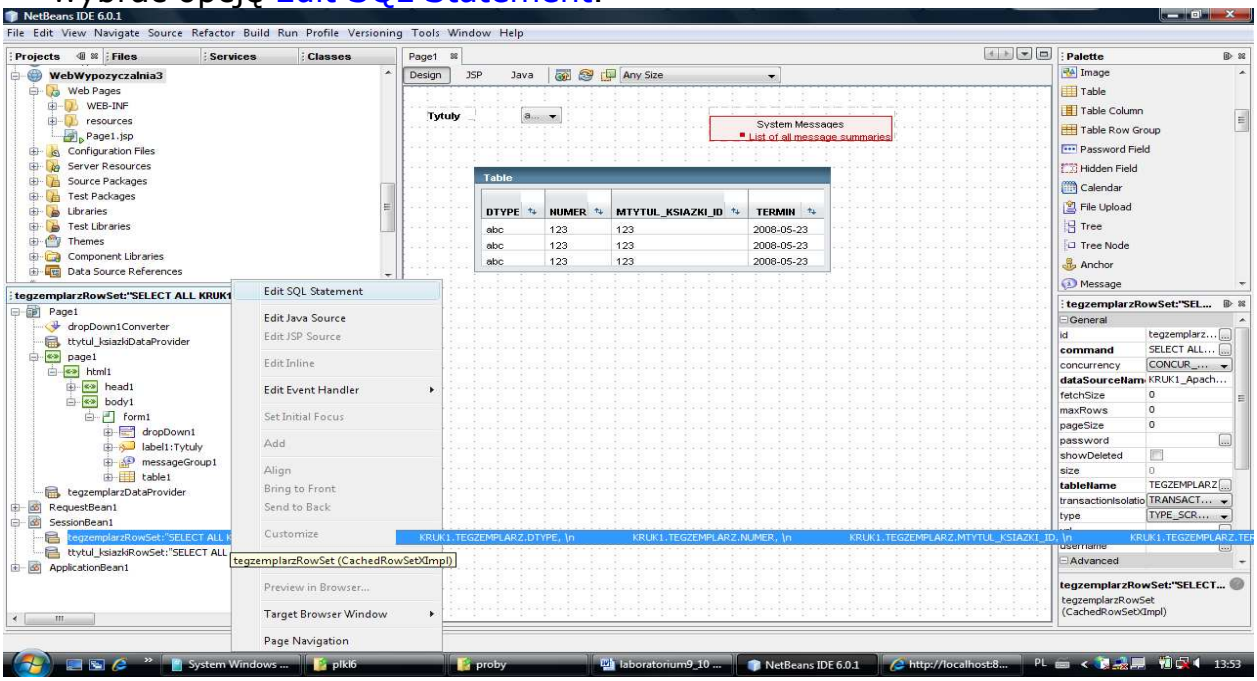
7.8. Kliknij prawym klawiszem na komponent table1 i wybierz Table Layout. W ukazanym formularzu należy wybrać kolumny tabeli do wyświetlenia. Za pomocą Ctrl+kliknięcie wybrać następujące kolumny: TEGZEMPLARZ.ID i nacisnąć przycisk < w celu usunięcia z widoku komponentu table1

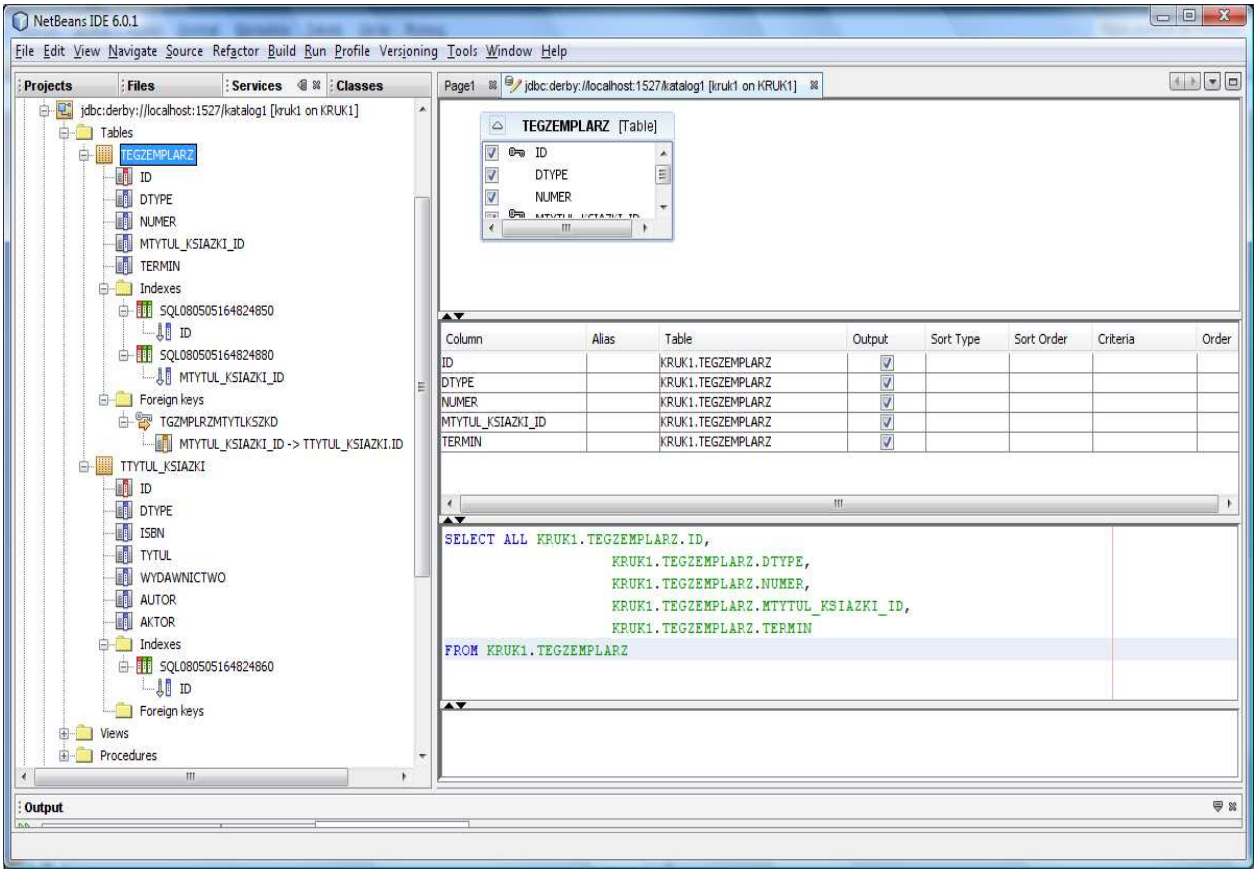
7.9. Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie Project na nazwę projektu, w ukazanym oknie uruchom kolejno Build Project, Deploy Project, Run Project lub tylko Run) i uruchom poszczególne funkcje aplikacji.)



7.10. Podgląd i możliwość edycji zapytania SQL dla warstwy bazodanowej tegzemplarzRowSet skojarzonej z SessionBean1

Należy wybrać w okienku Navigator komponent SessionBean1 i wybrać tegzemplarzRowSet prawym klawiszem myszy. Z wyskakującego menu wybrać opcję Edit SQL Statement.

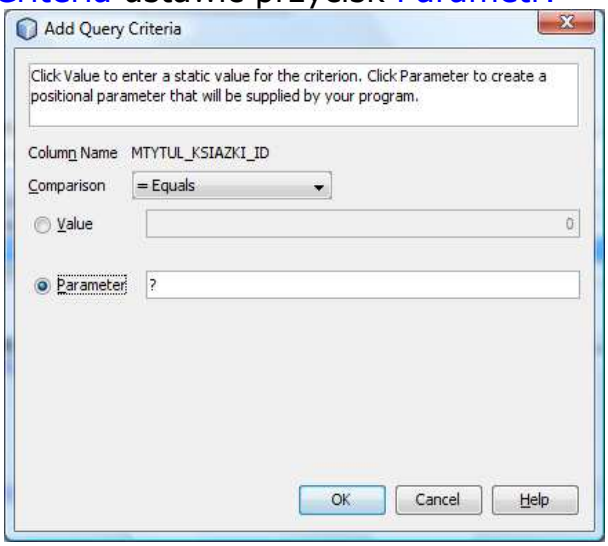




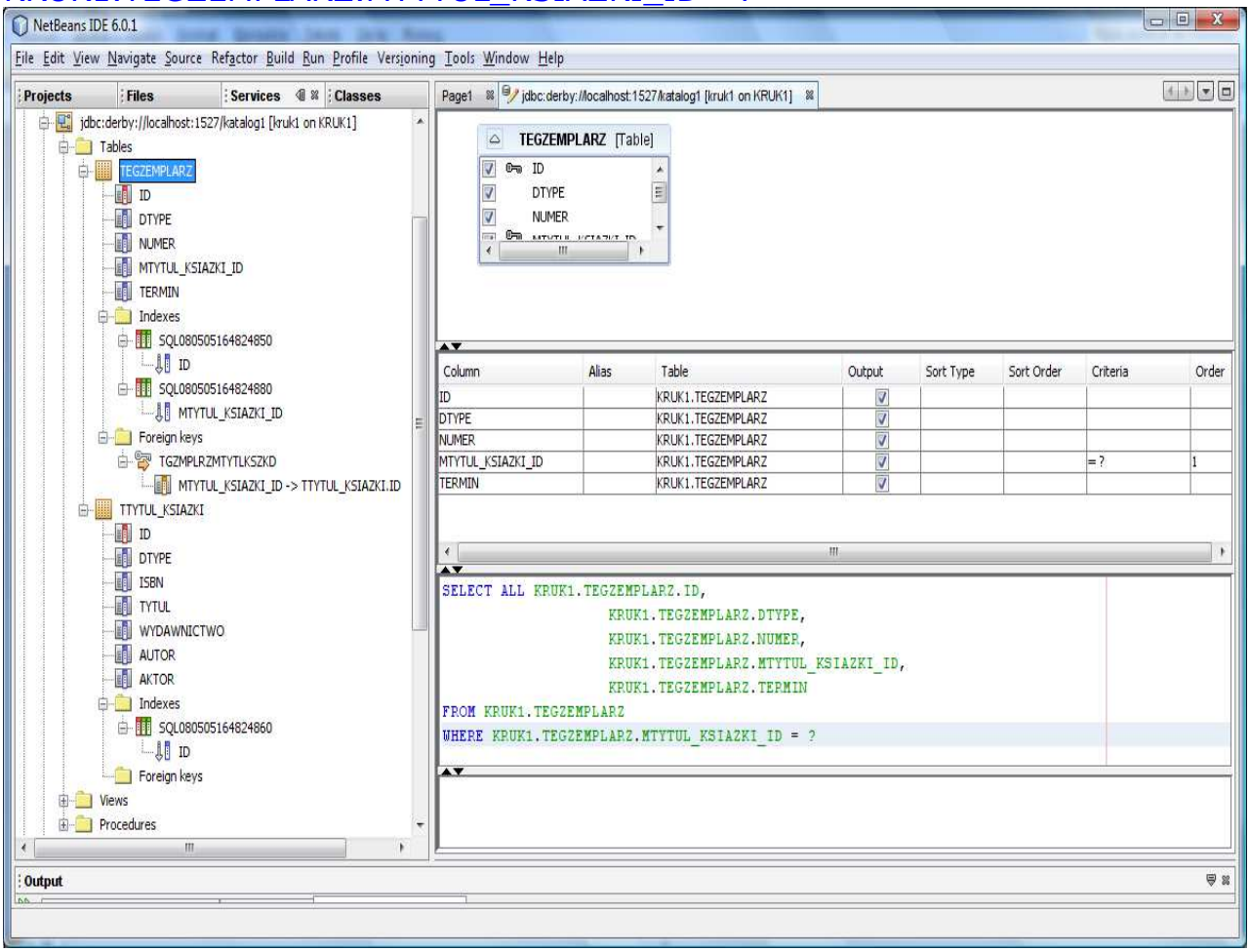
7.11. Ograniczenie wyświetlanych wierszy w tabeli do tytułu książki wybranej w komponencie downDrop1.

Uwaga: W tym rozwiązaniu kryterium dla zapytania Select jest kolumna MTYTUL_KSIAZKI_ID

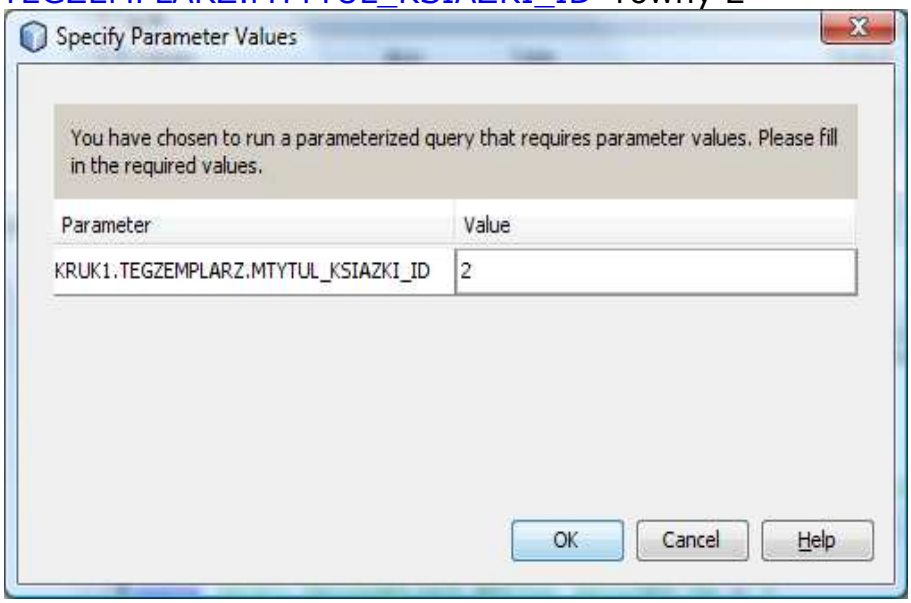
- W kolumnie MTYTUL_KSIAZKI_ID kliknąć prawym klawiszem myszy i wybrać Add Query Criteria
- W menu Add Query Criteria ustawić przycisk Parametr.



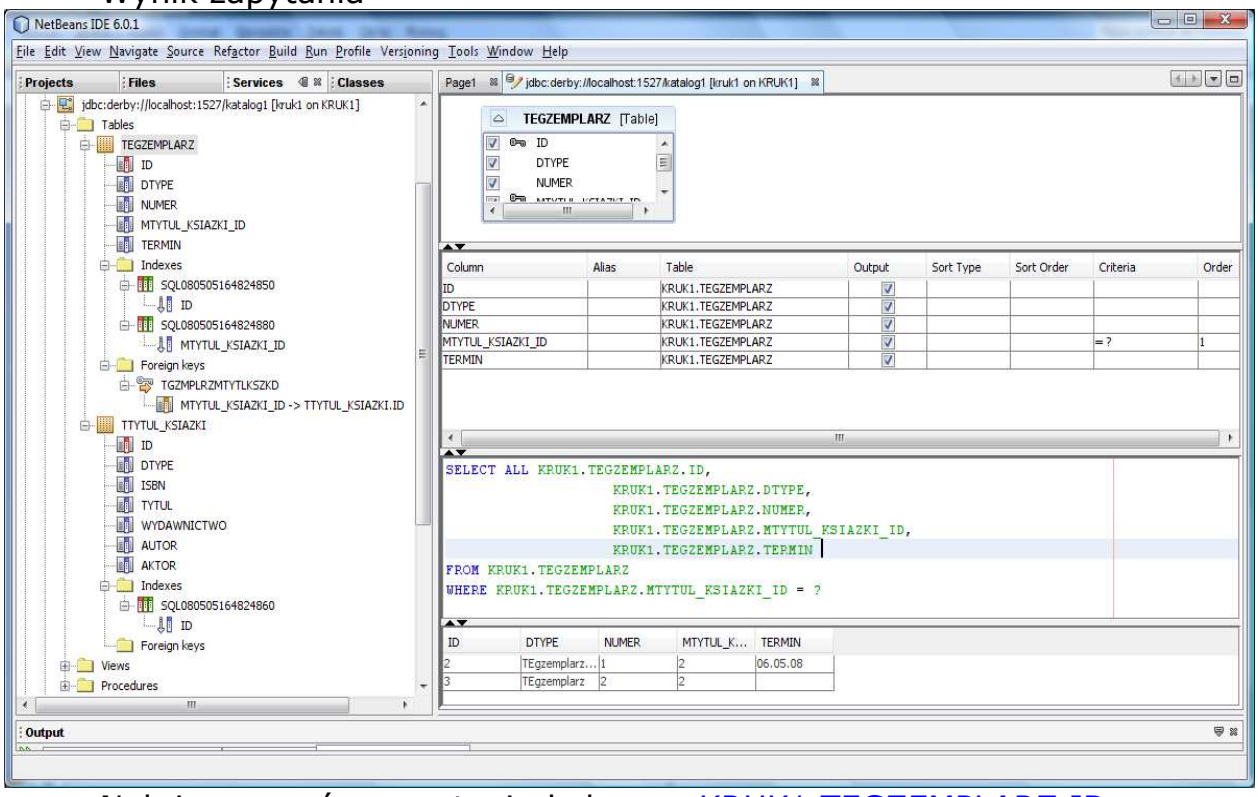
- Zapytanie zostanie zmodyfikowane o warunek WHERE KRUK1.TEGZEMPLARZ.MTYTUL_KSIAZKI_ID =?



- Można sprawdzić działanie zapytania klikając prawym klawiszem myszy na powierzchnię edytora i wybierając opcje Run Query.
- Podanie parametru zapytania np. KRUK1.TEGZEMPLARZ.MTYTUL_KSIAZKI_ID równy 2

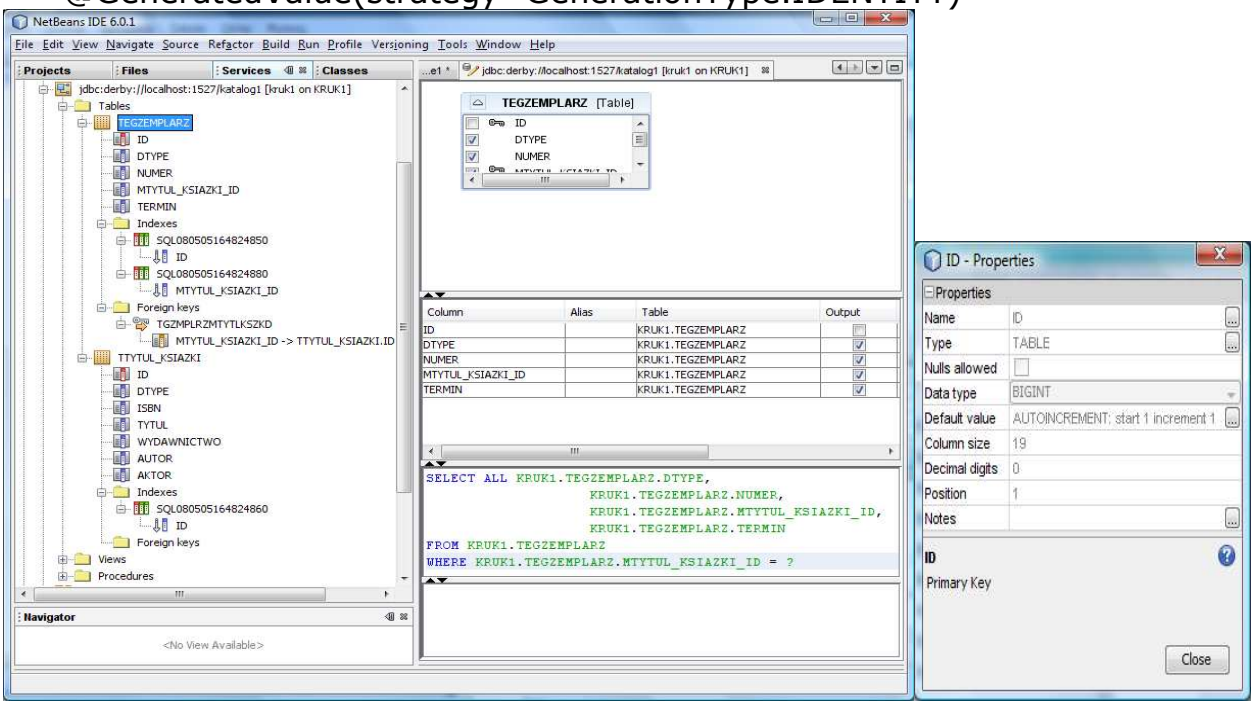


- Wynik zapytania

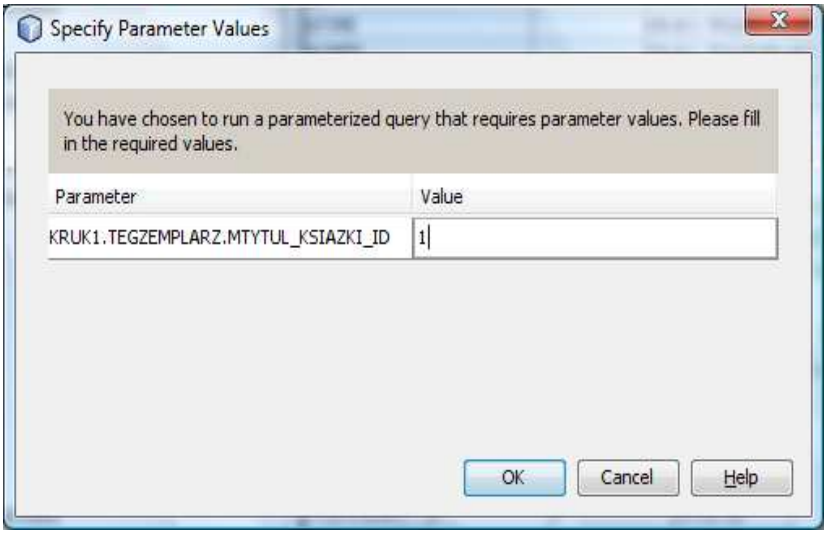


- Należy usunąć z zapytania kolumnę KRUK1.TEGZEMPLARZ.ID – pozwoli to wykorzystać komponent Table również do wstawiania danych do tabeli TEGZEMPLARZ bez naruszania więzów integralności w zakresie przydzielania kluczy głównych krotkom tabeli – klucze te przydziela baza danych (Default value = AUTOINCREMENT), natomiast pozostawienie kolumny Id powodowałoby wysyłanie wartości klucza głównego z aplikacji do bazy danych podczas wstawiania danych, co pozostawałoby w kolizji ze strategią generowania kluczy głównych. Zostało to ustalone przy definiowaniu tabel za pomocą adnotacji typu Entity:

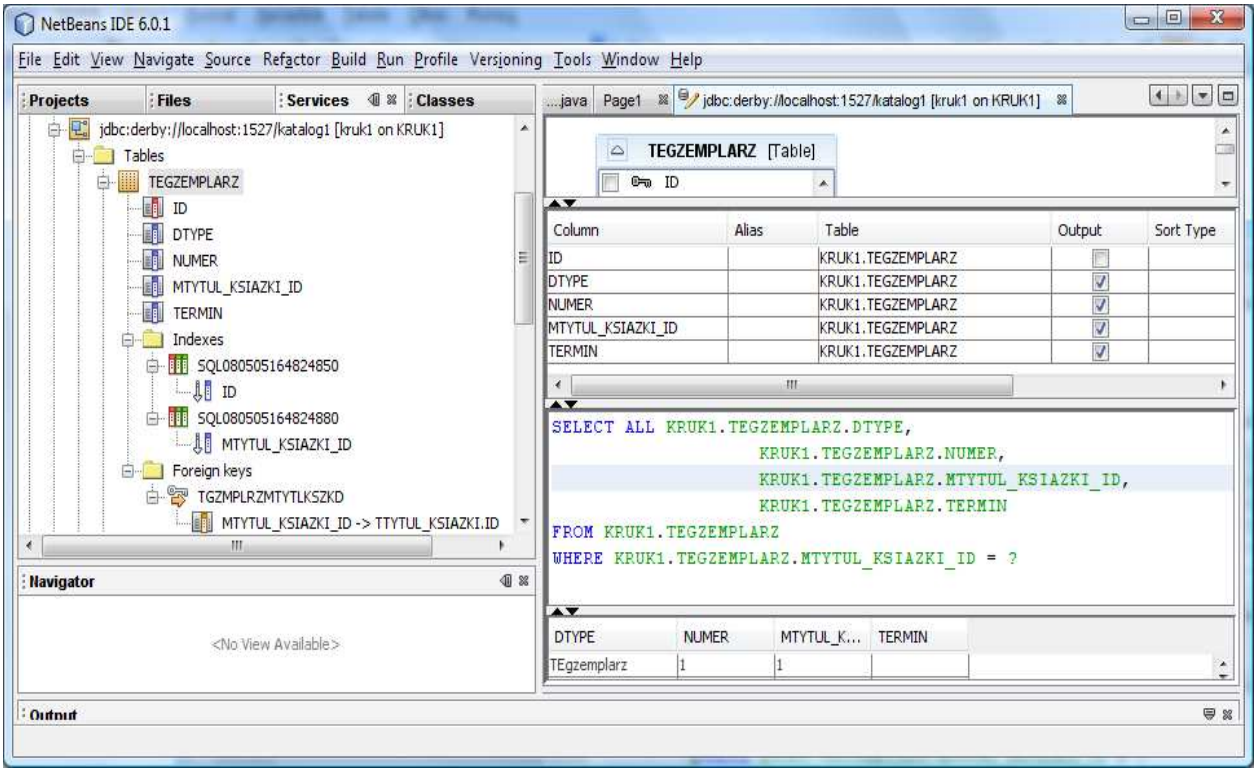
```
@Id
@GeneratedValue(strategy=GenerationType.IDENTITY)
```



- Można sprawdzić działanie zapytania klikając prawym klawiszem myszy na powierzchnię edytora i wybierając opcje Run Query.
- Podanie parametru zapytania np. KRUK1.TEGZEMPLARZ.MTYTUL_KSIAZKI_ID równy 1



- Wynik zapytania

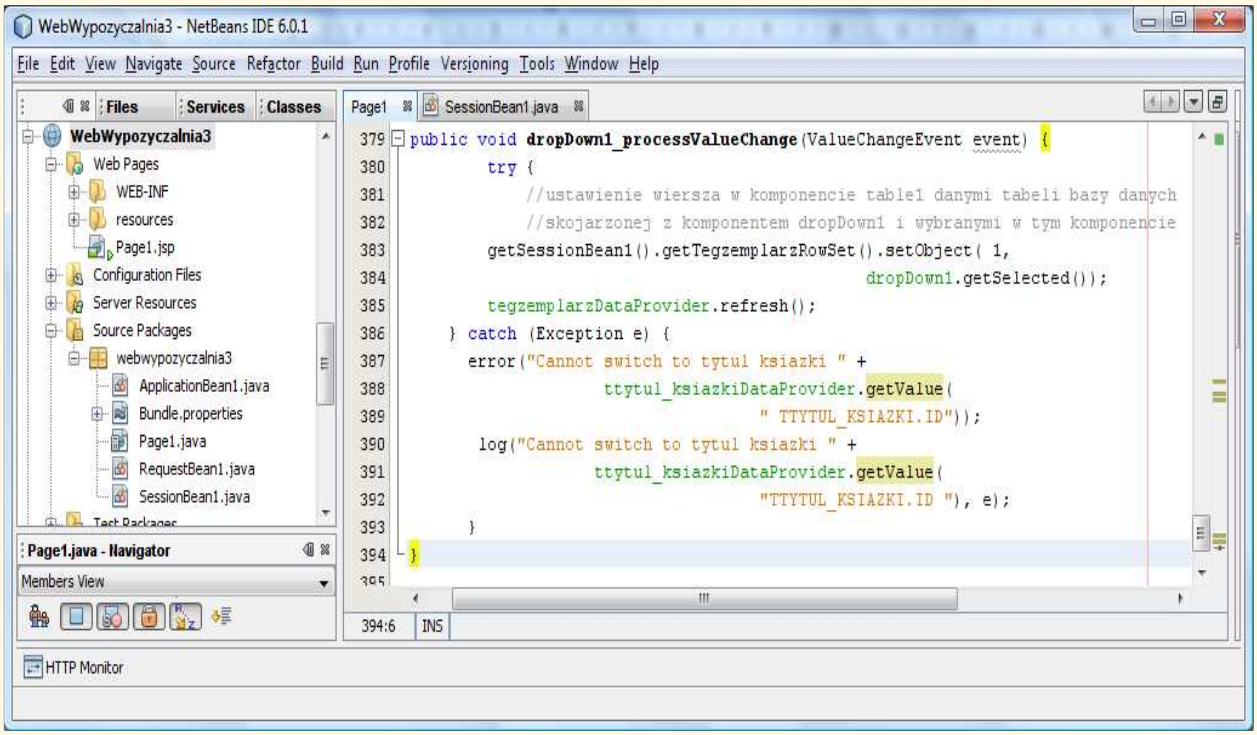


Zamknij formularz klikając na klawisz ”X” zakładki tegzemplarzRowSet(SessionBean1)

8. Uzupełnij kod aplikacji w trybie Java strony Page1 (klasa Page1 dziedziczy od klasy AbstractPageBean)

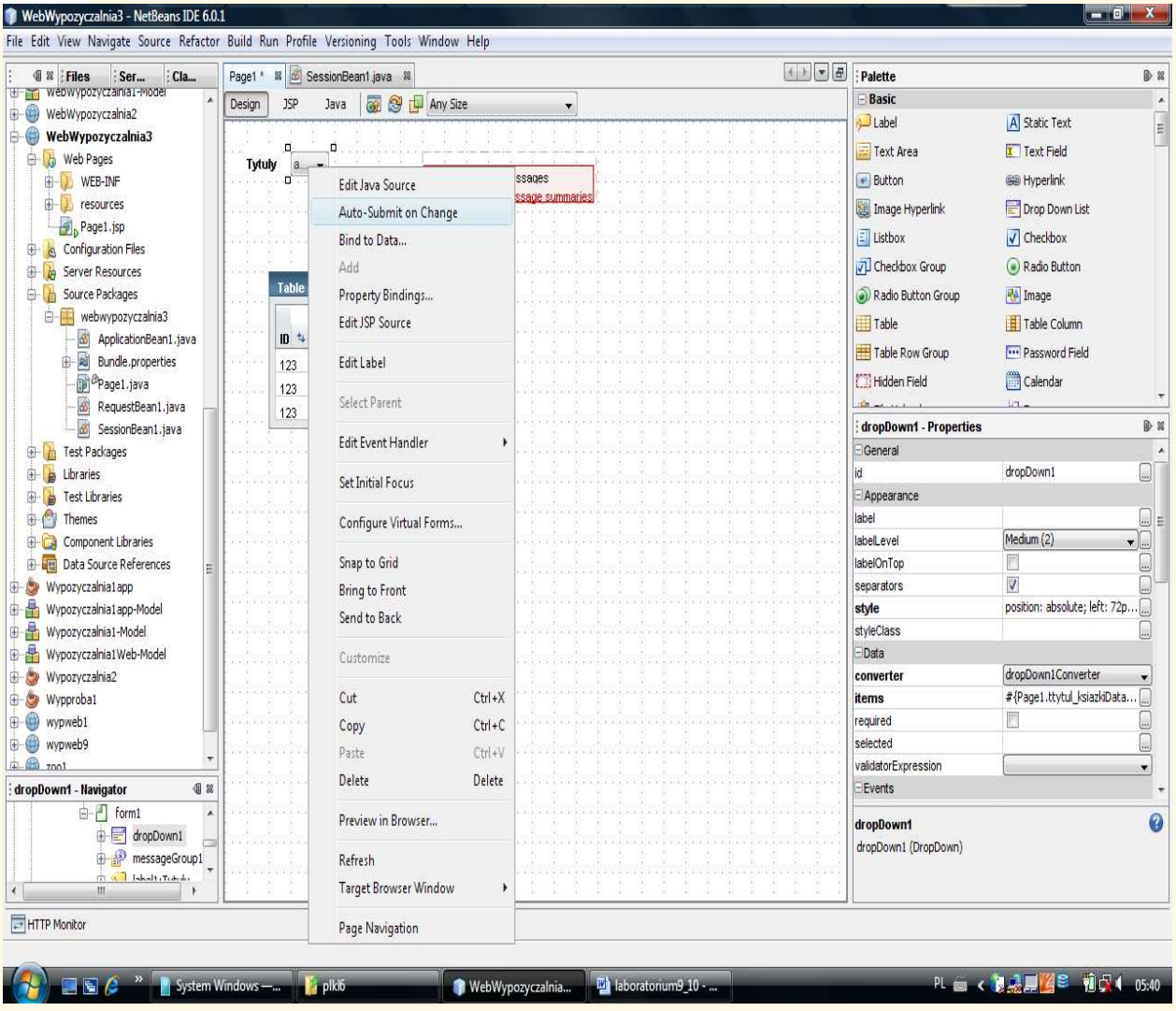
8.1. Należy kliknąć podwójnie na kontrolkę dropDown1 i obsłużyć zdarzenie ProcessValueChange za pomocą metody dropDown1_processValueChange

```
public void dropDown1_processValueChange(ValueChangeEvent event)
{
  try { //ustawienie wiersza w komponencie table1 danymi tabeli bazy danych skojarzonej z komponentem dropDown1 i
        //wybranymi w tym komponencie
          getSessionBean1().getTegzemplarzRowSet().setObject(
                                                  1, dropDown1.getSelected());
         tegzemplarzDataProvider.refresh();
      } catch (Exception e) {
        error("Cannot switch to tytul ksiazki " +
                ttytul_ksiazkiDataProvider.getValue("TYTUL_KSIAZKI.ID"));
        log("Cannot switch to tytul ksiazki " +
           ttytul_ksiazkiDataProvider.getValue("TTYTUL_KSIAZKI.ID "), e);
         }
}
```



8.2. Przejdź do trybu Design strony Page1, zaznacz komponent dropDown1 i prawym klawiszem myszy kliknij nad tym zaznaczonym komponentem. Z wyskakującego menu ustaw Auto-Submit on Change.

Pozwoli to obsługiwać zdarzenia wyboru pozycji z listy rozwijanej komponentu dropDown1, typu ProcessValueChange, za pomocą metody dropDown1_processValueChange. Zdarzeni te mają charakter typu Submit (przesłania informacji z formularza klienta do serwera WWW i serwera aplikacji)

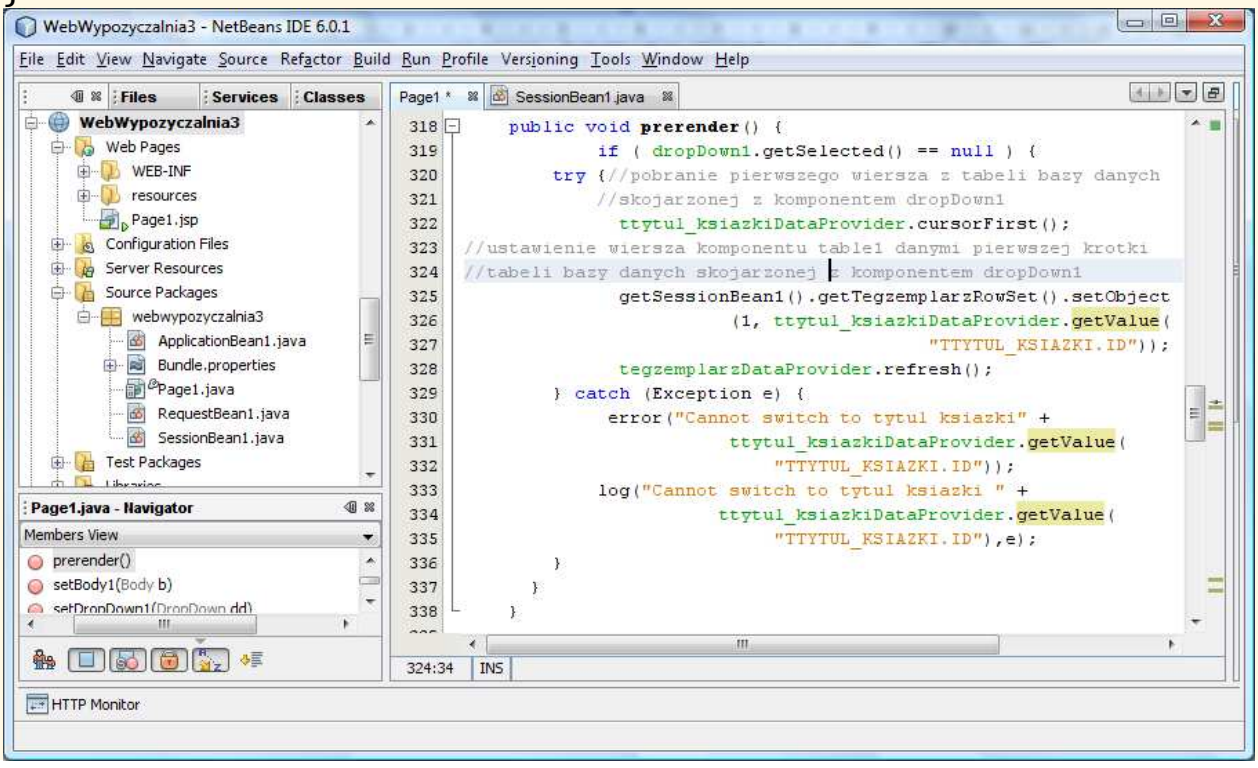


Wstawienie przycisku Dodaj egzemplarz

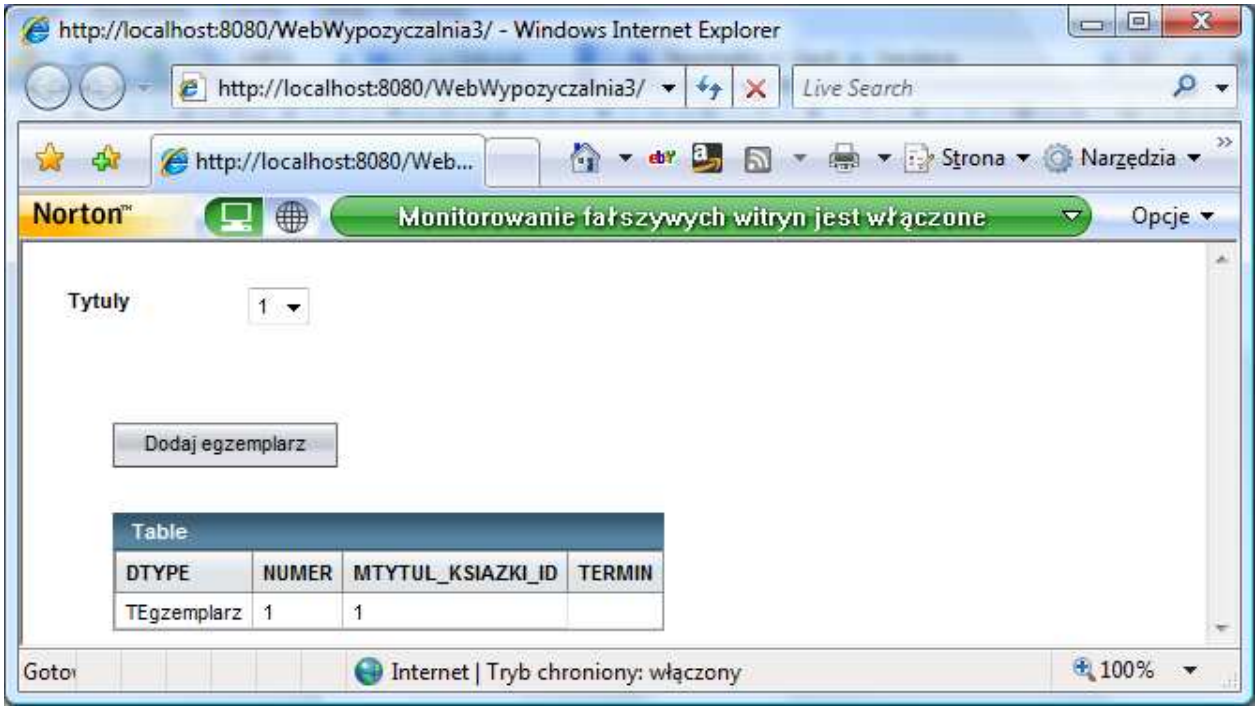
- Należy przeciągnąć komponent typu Button i napisać na nim Dodaj egzemplarz.

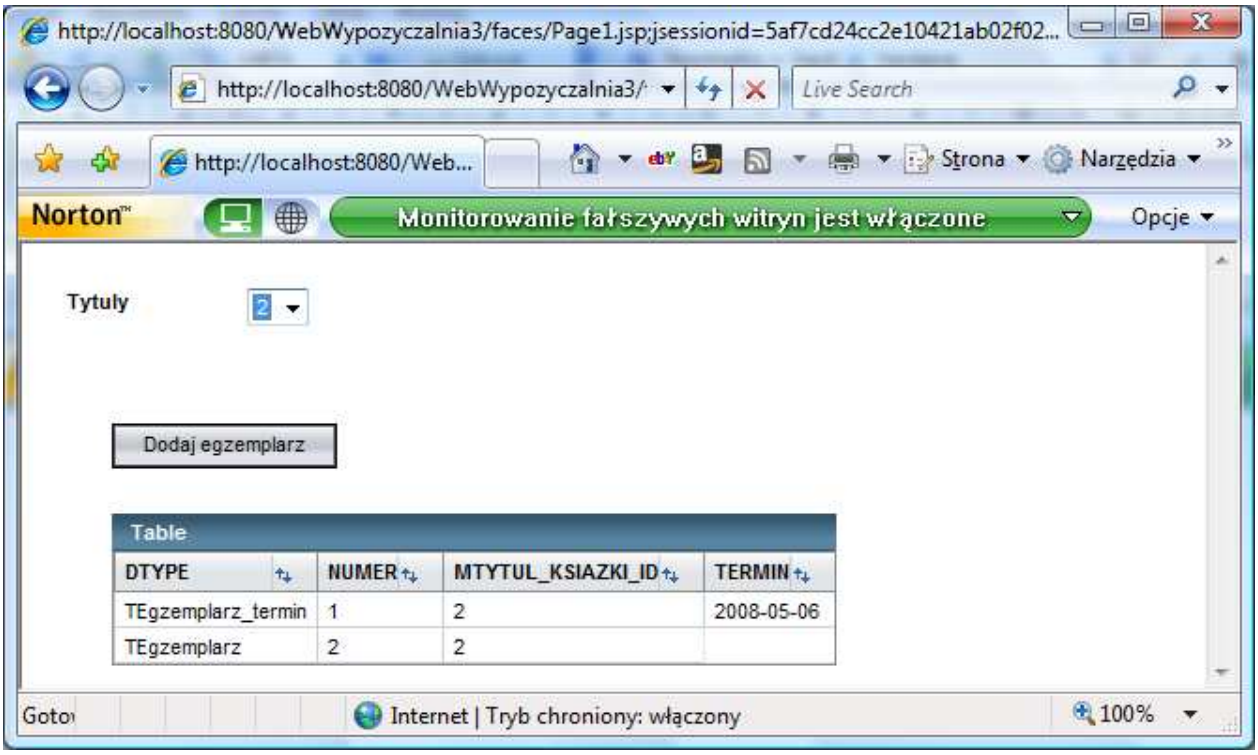
8.3. metoda prerender() - inicjowanie zawartości komponentu table1 typu Table podczas odświeżania strony. Jest to konieczne, gdyż w przeciwnym wypadku zawartość komponentu typu Table byłaby nieokreślona podczas wyświetlania strony, co powoduje błędy wykonania aplikacji – ponieważ zawartość tego komponentu jest uzależniona od wybranej krotki z tabeli TTYTUL_KSIAZKI.

```
public void prerender()
 {
   if ( dropDown1.getSelected() == null )
   { try
     {
//pobranie pierwszego wiersza z tabeli bazy danych skojarzonej z komponentem dropDown1
       ttytul_ksiazkiDataProvider.cursorFirst();
//ustawienie wiersza komponentu table1 danymi pierwszej krotki tabeli bazy danych skojarzonej z komponentem dropDown1
       getSessionBean1().getTegzemplarzRowSet().setObject
           (1, ttytul_ksiazkiDataProvider.getValue("TTYTUL_KSIAZKI.ID"));
       tegzemplarzDataProvider.refresh();
     } catch (Exception e)
       { error("Cannot switch to tytul ksiazki" +
               ttytul_ksiazkiDataProvider.getValue("TTYTUL_KSIAZKI.ID"));
         log("Cannot switch to tytul ksiazki " +
               ttytul_ksiazkiDataProvider.getValue("TTYTUL_KSIAZKI.ID"),e);
       }
   }
}
```



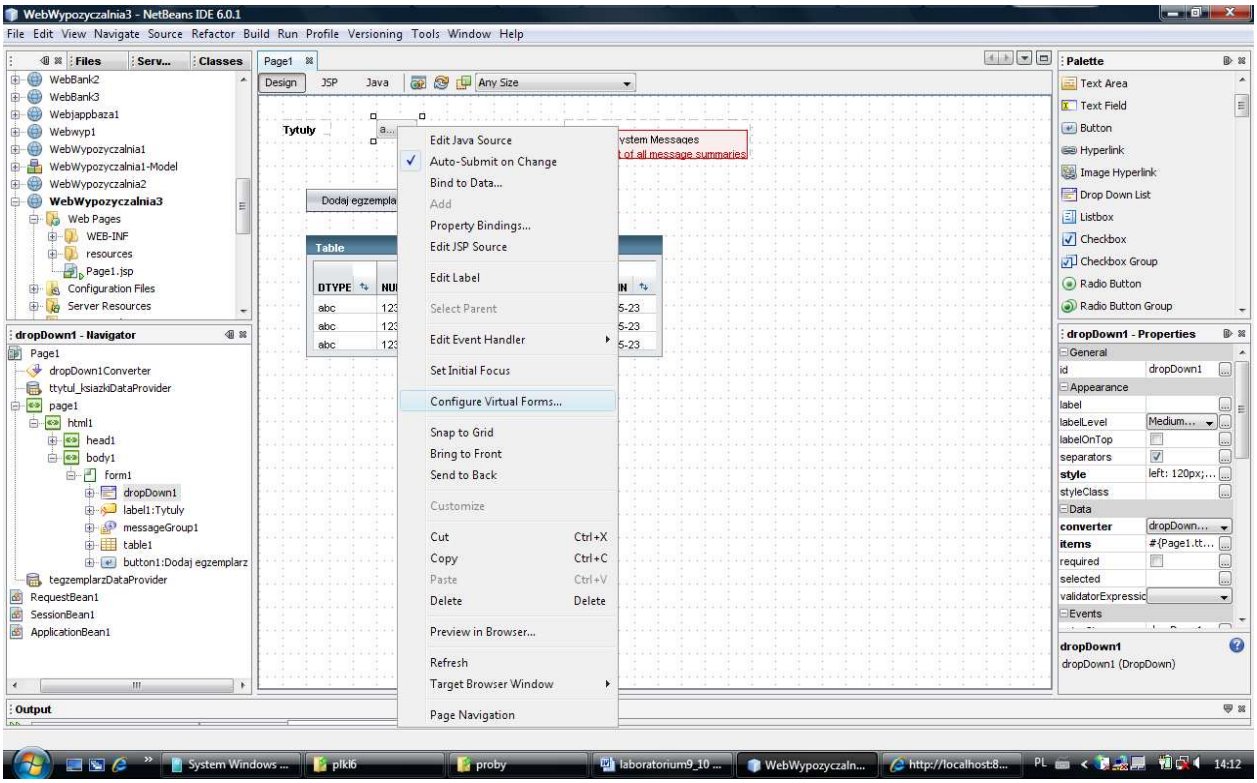
8.3. Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie Project na nazwę projektu, w ukazanym oknie uruchom kolejno Build Project, Deploy Project, Run Project lub tylko Run Project) i uruchom poszczególne funkcje aplikacji.



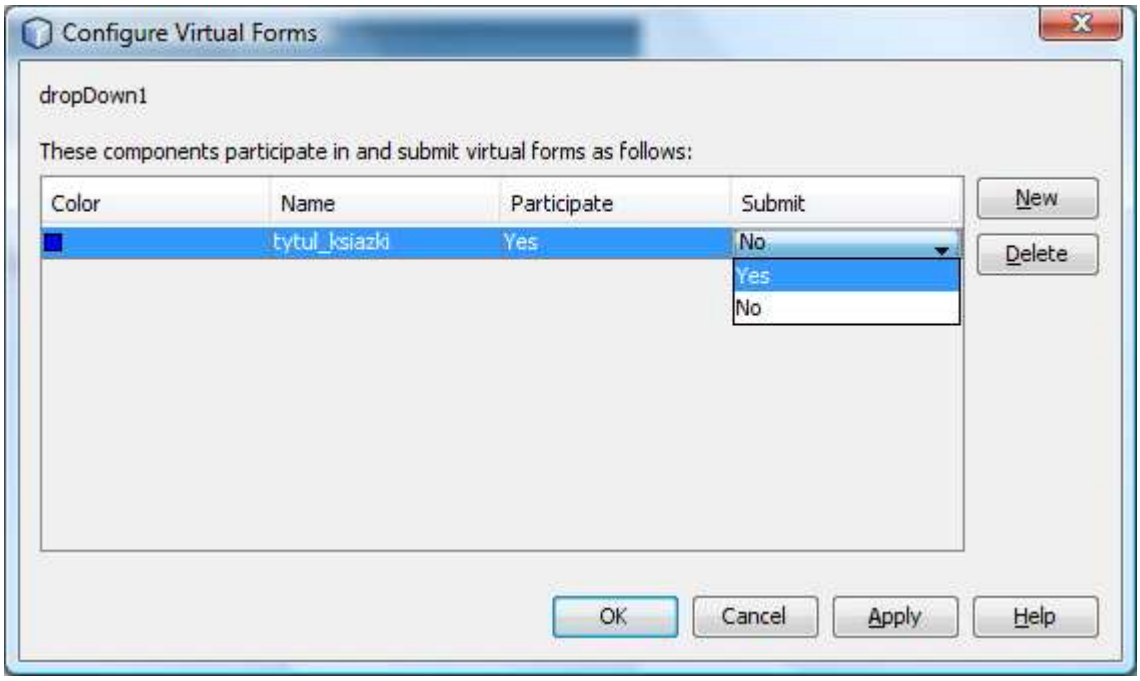


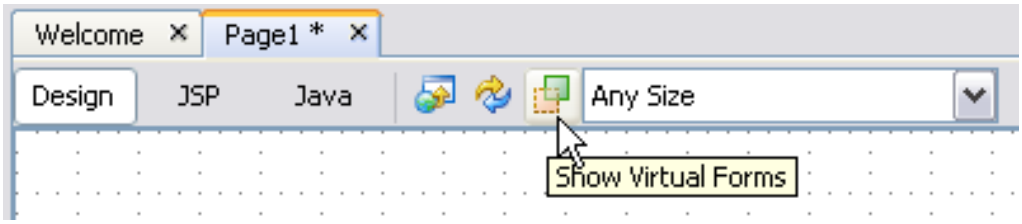
9. Konfiguracja wirualnego formularza (Configure Virtual Forms) – **wirtualny formularz umożliwia eliminowanie konwersji i walidacji w przesłanych danych do serwerów www i aplikacji, należących do wybranego wirtualnego formularza.**

9.1. W trybie Design strony Page1 należy wybrać komponent dropDown1 i kliknąć prawym klawiszem myszy. Z wyskakującego menu wybrać Configure Virtual Forms.

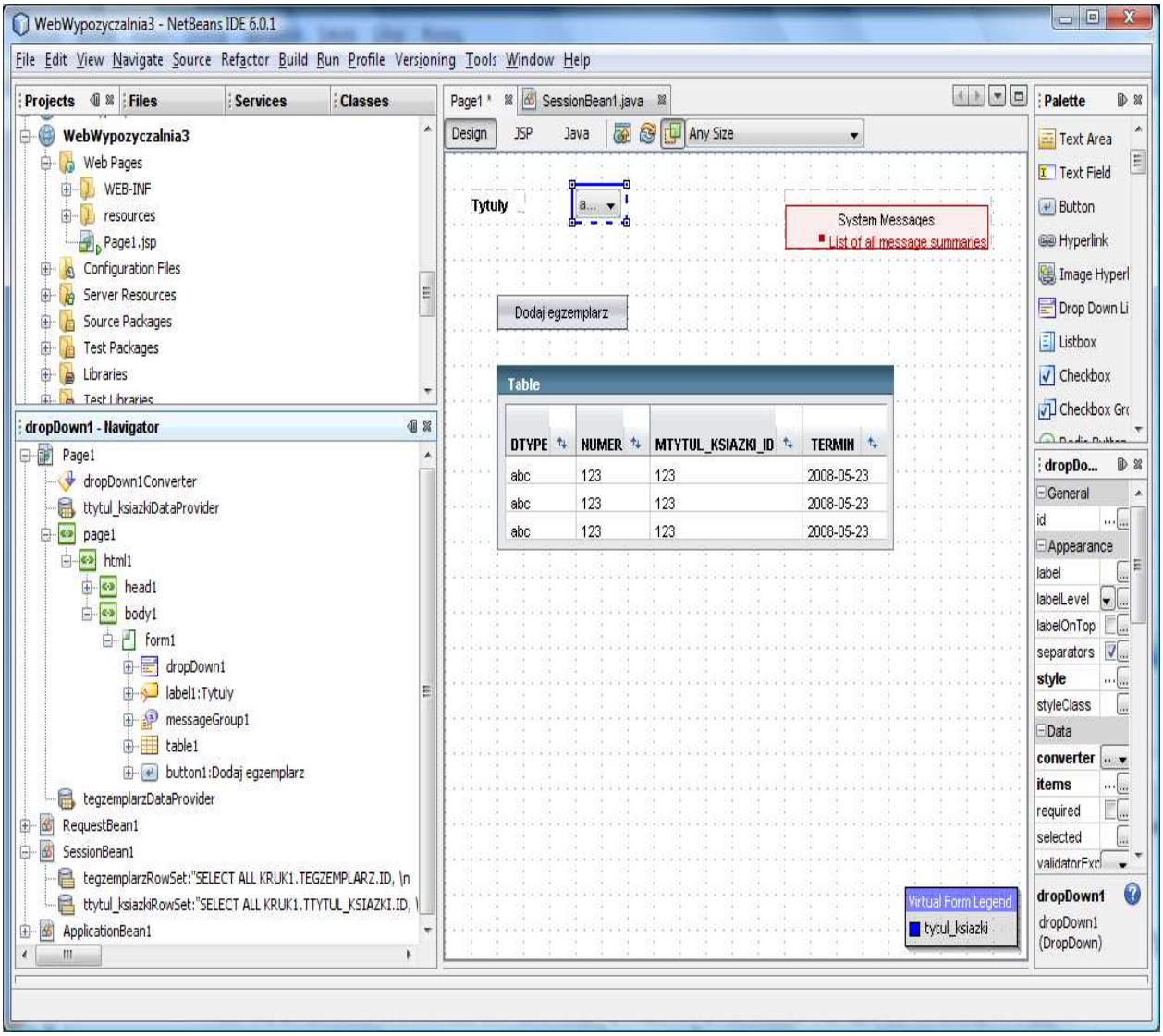


9.2. W formularzu opcji nacisnąć klawisz New. Nowy formularz nazwać np. tytul_ksiazki i oraz kliknąć dwa razy w kolumny Participate i Submit i w obu ustawić Yes (wybór pozycji z listy i wykonanie operacji Submit dla strony www). Zatwierdzić ustawienia klawiszem OK.





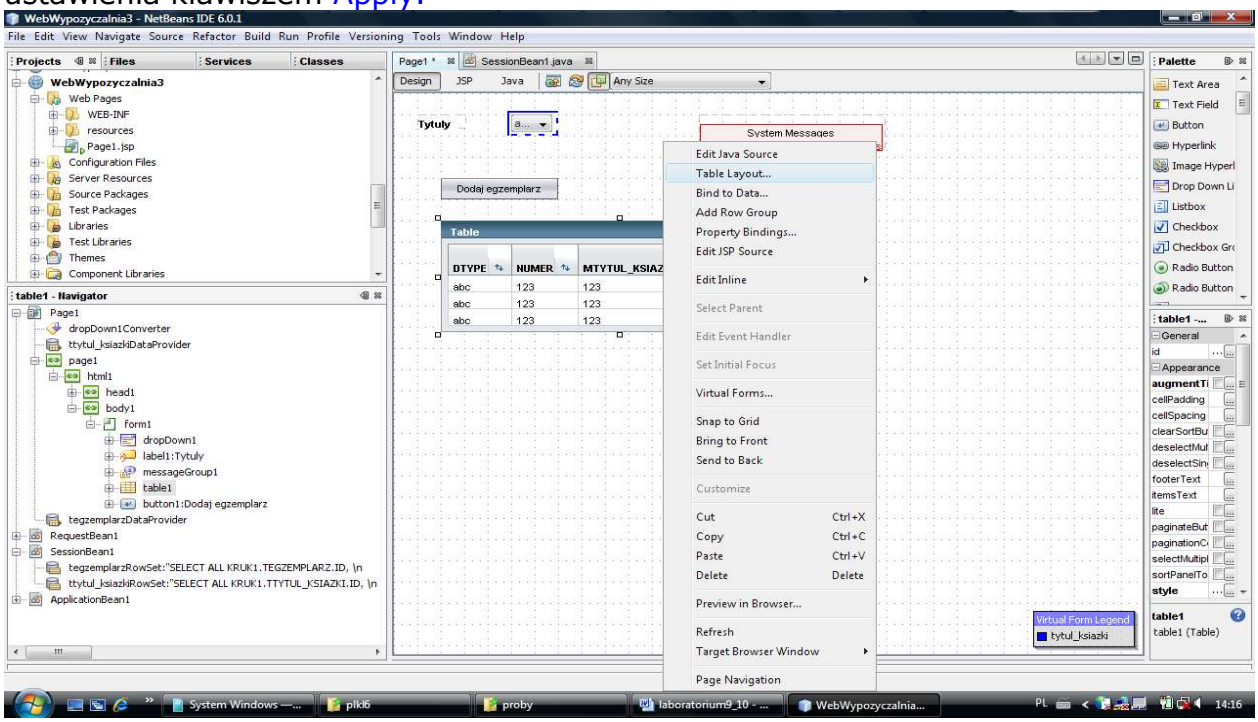
W wyniku ustawienia wirtualnego formularza tytul_ksiazki nie są sprawdzane dane wyświetlane w komponencie table1. W trybie Design komponent należący do wirtualnego formularza jest podświetlony, pod warunkiem, że zostanie on pokazany po kliknięciu na ikonę pokazaną powyżej na belce narzędziowej.



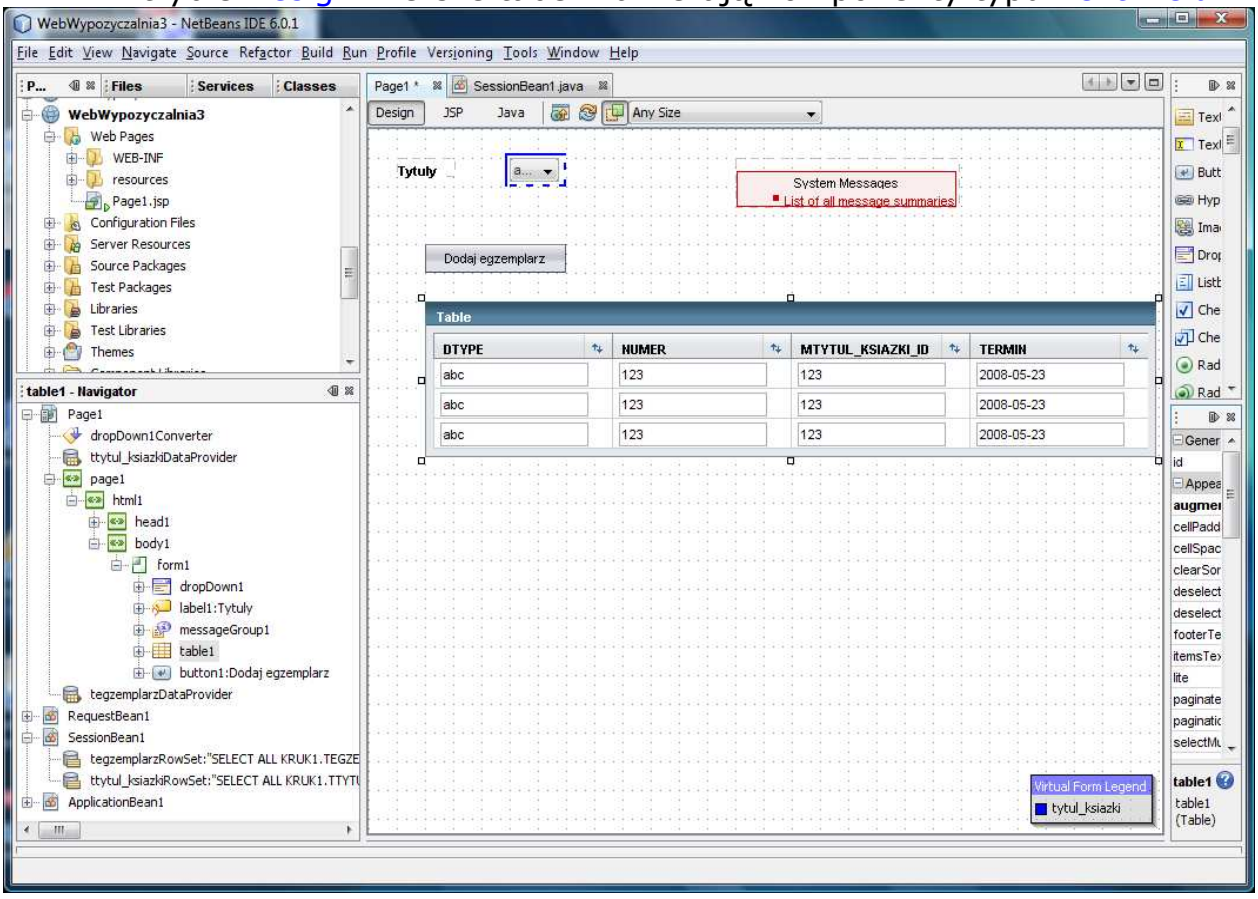
9.3. Wstawianie komponentów wejściowych do komórek komponentu table1

- Po zaznaczeniu komponentu table1 należy kliknąć prawym klawiszem myszy i wybrać z wyskakującego menu wybrać Table Layout.

W ukazanym formularzu wybrać kolejno kolumny tabeli w prawej części formularza i ustawić typ komponentu Text Field w liście Komponent Type i zatwierdzić kolejne ustawienia klawiszem Apply.

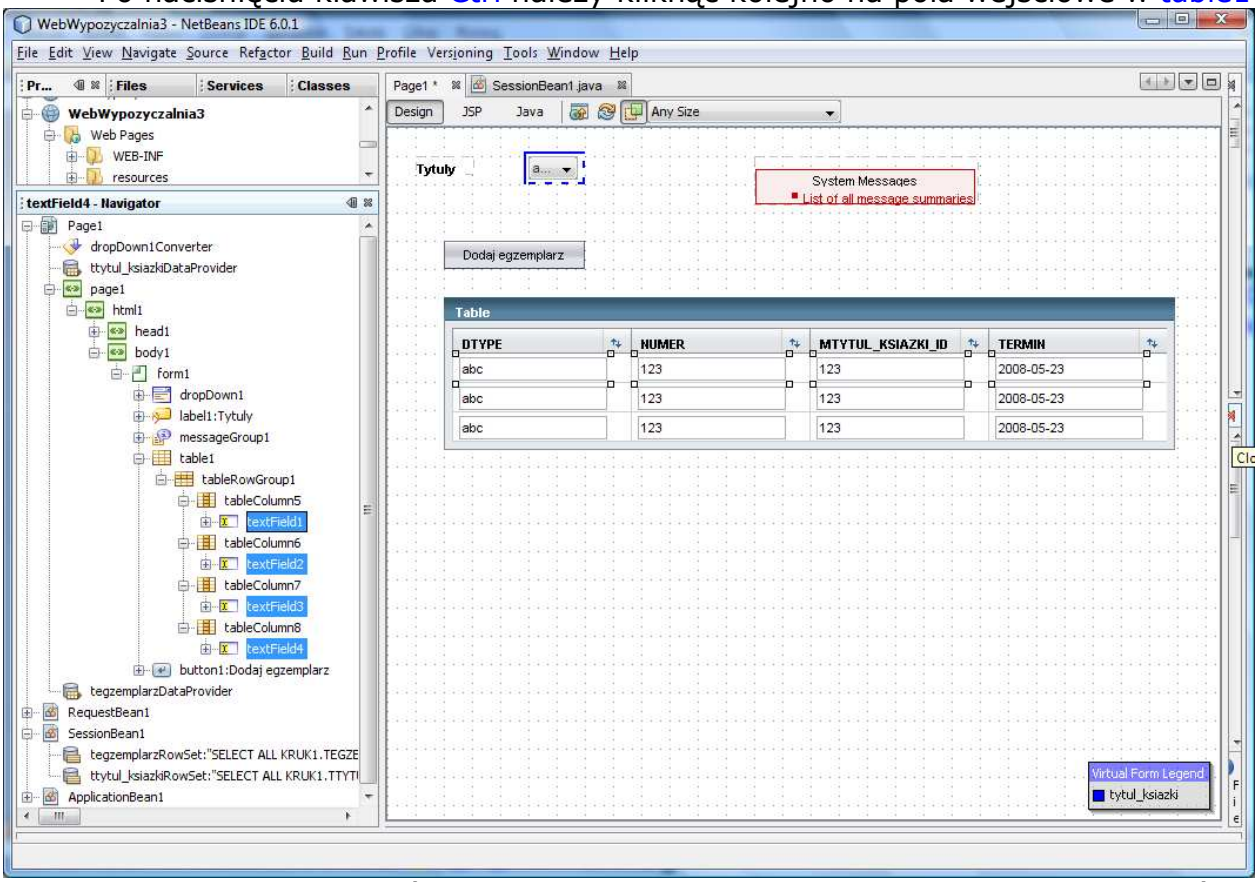


- W trybie Design wiersze tabeli zawierają komponenty typu Text Field

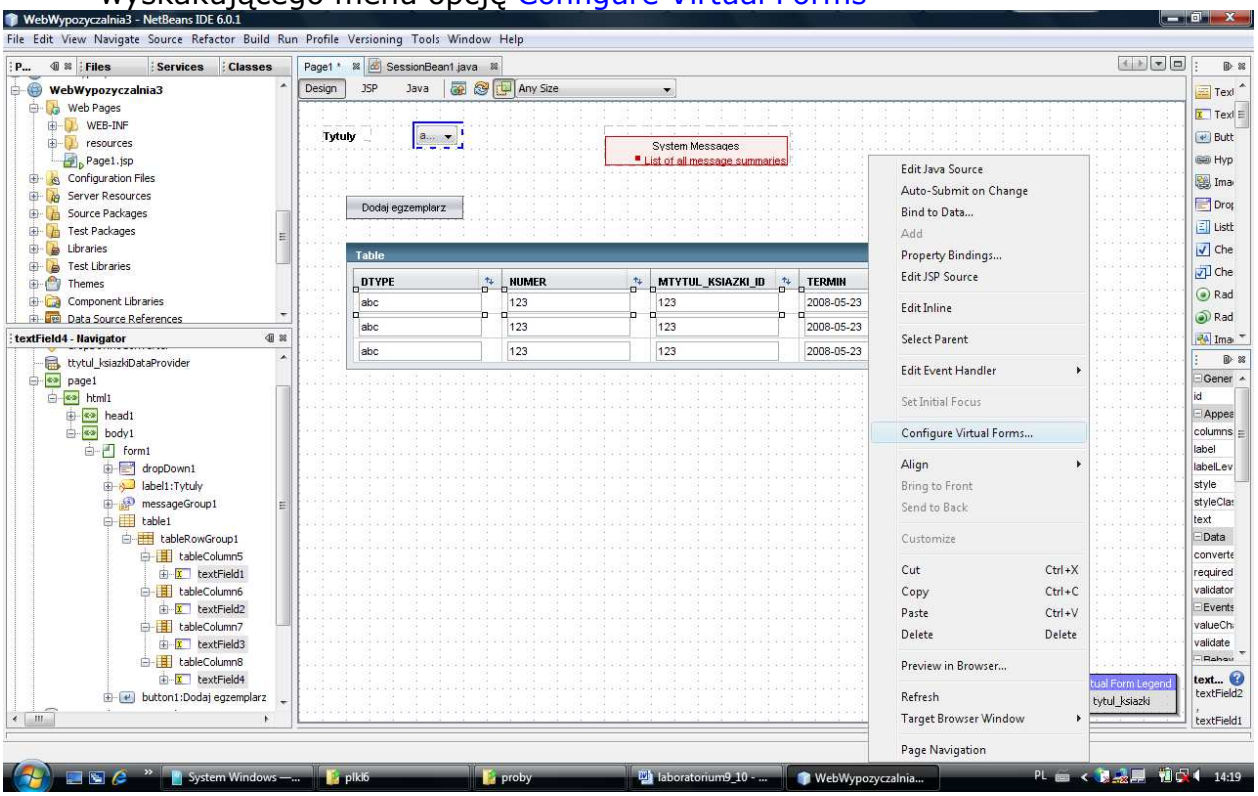


9.4. Utworzenie wirtualnego formularza dla pól wejściowych w komponencie table1

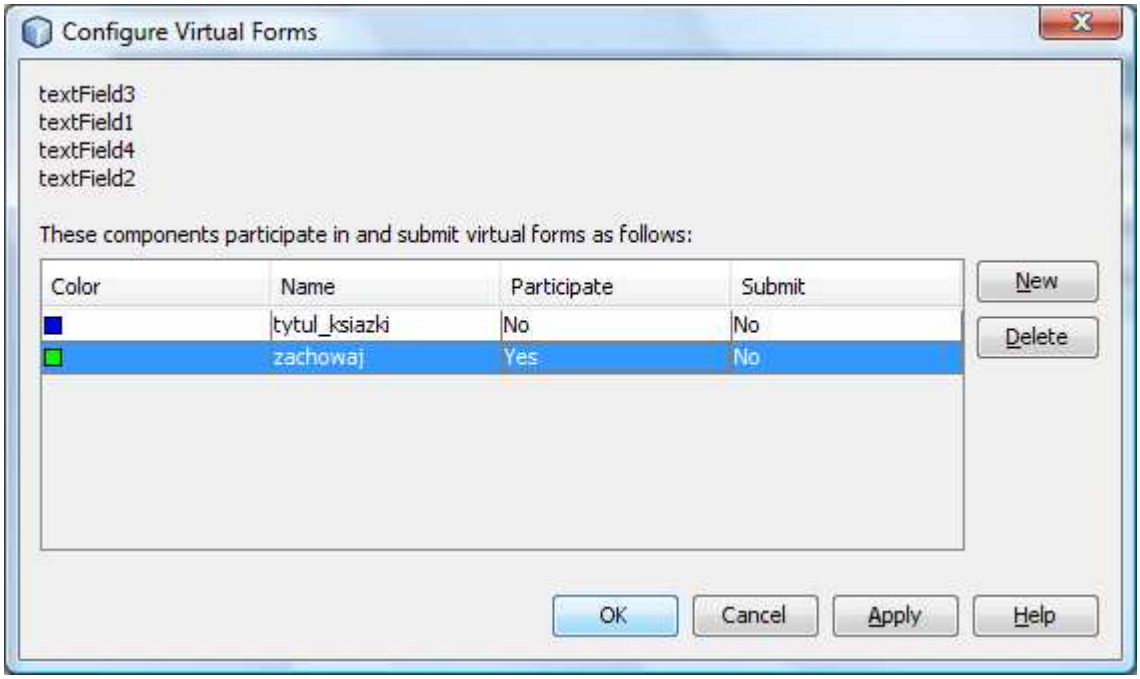
- Po naciśnięciu klawisza Ctrl należy kliknąć kolejno na pola wejściowe w table1



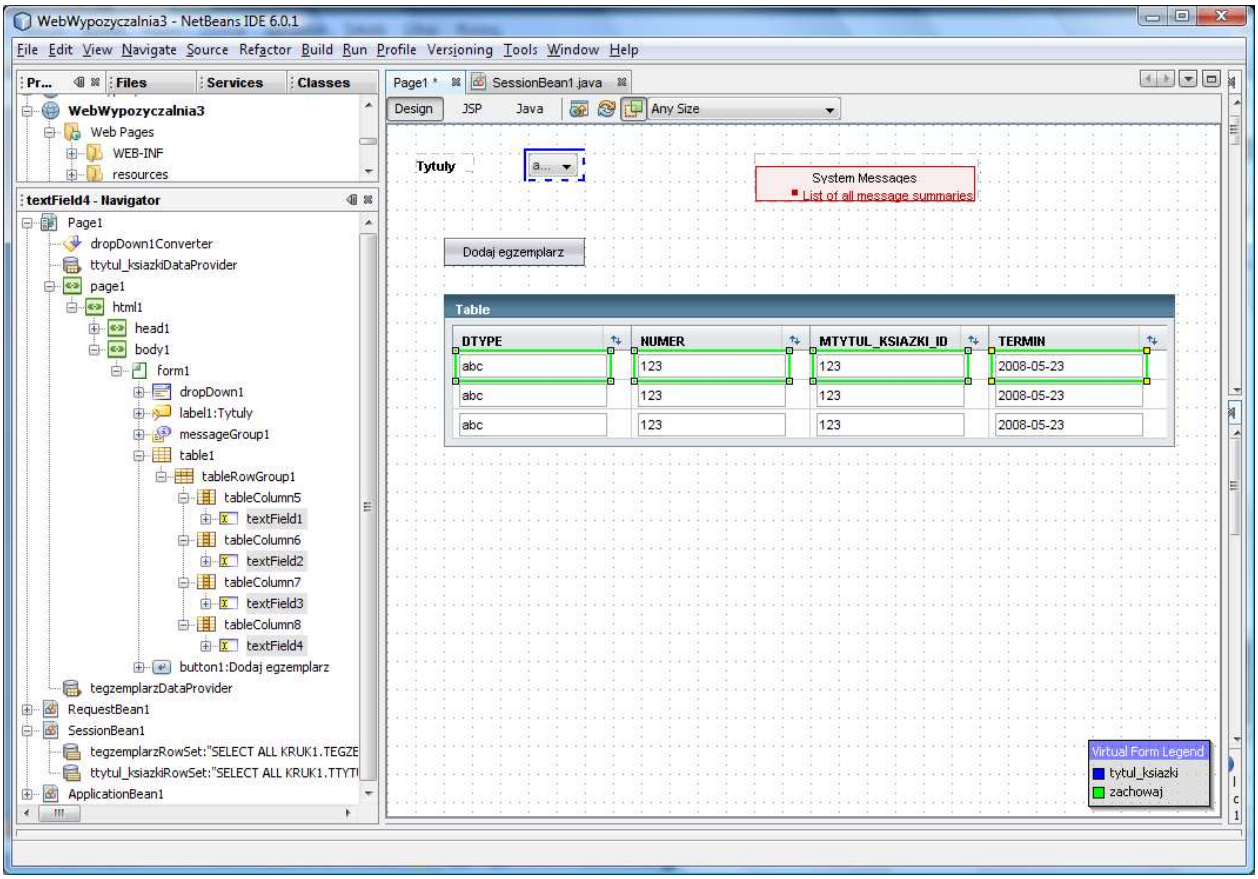
- Następnie po naciśnięciu prawego klawisza myszy należy wybrać z wyskakującego menu opcję Configure Virtual Forms



- Po naciśnięciu klawisza New należy wstawić nowy formularz np. o nazwie zachowaj. Należy podwójnie kliknąć dwukrotnie kolumnę Participate i ustawić Yes i zatwierdzić nowy formularz klawiszem OK.



- Na stronie Page1 w trybie Design zaznaczone są odrębnymi kolorami komponenty należące do tego samego wirtualnego formularza

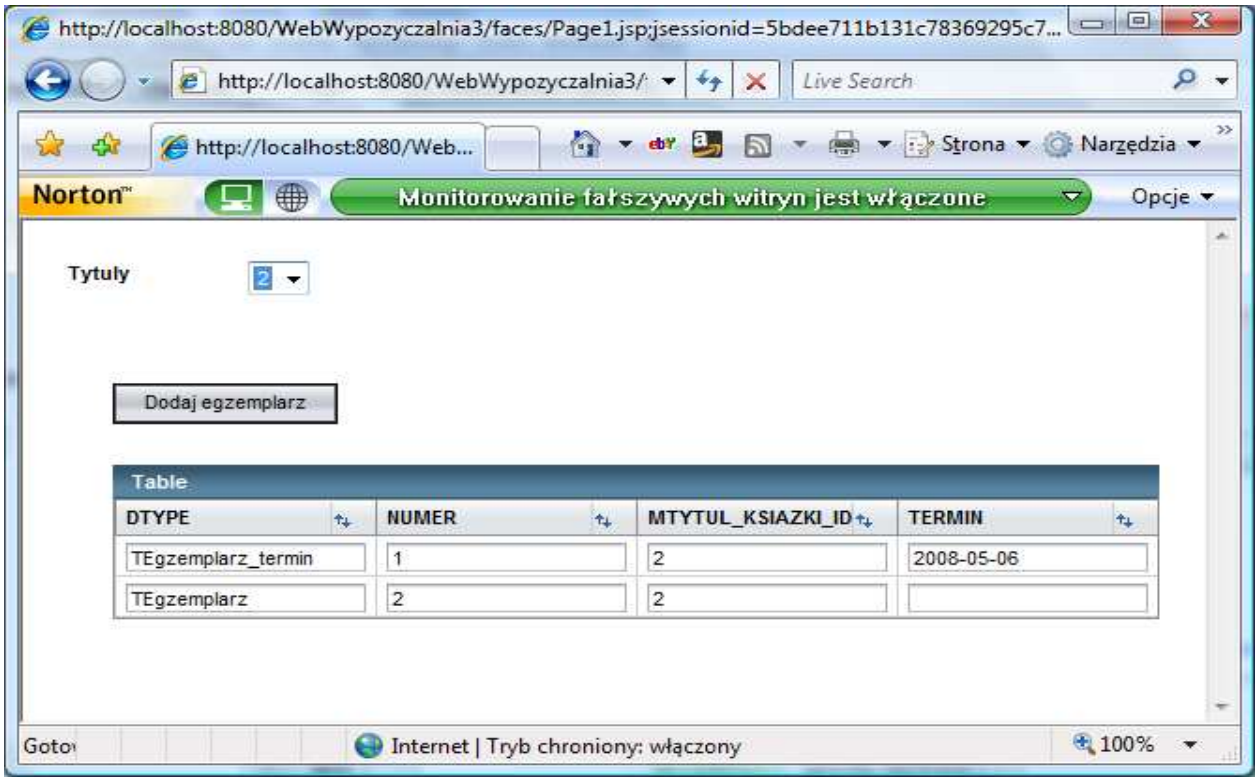


9.5. Uzupełnienie kodu źródłowego metody dropDown1_processValueChange dla zdarzenia typu ProcessValueChange

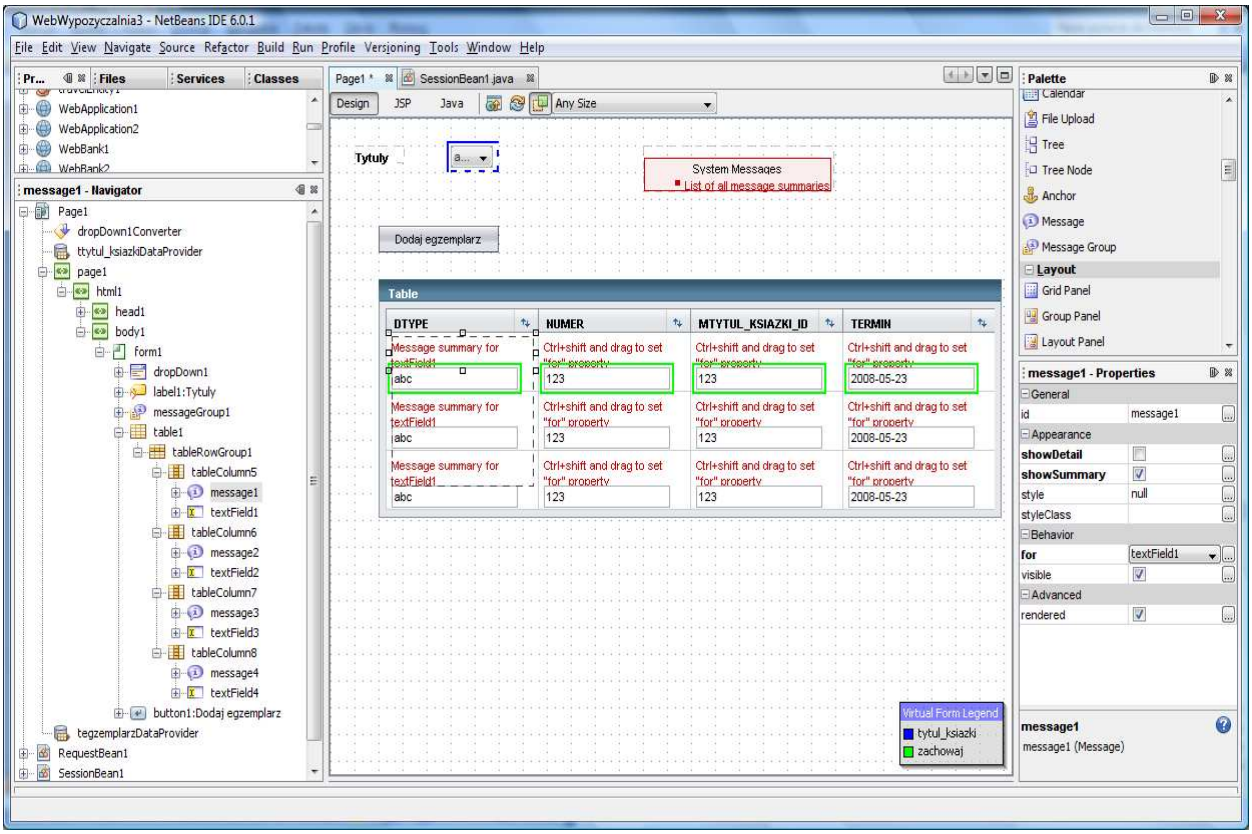
Plik Page1.java

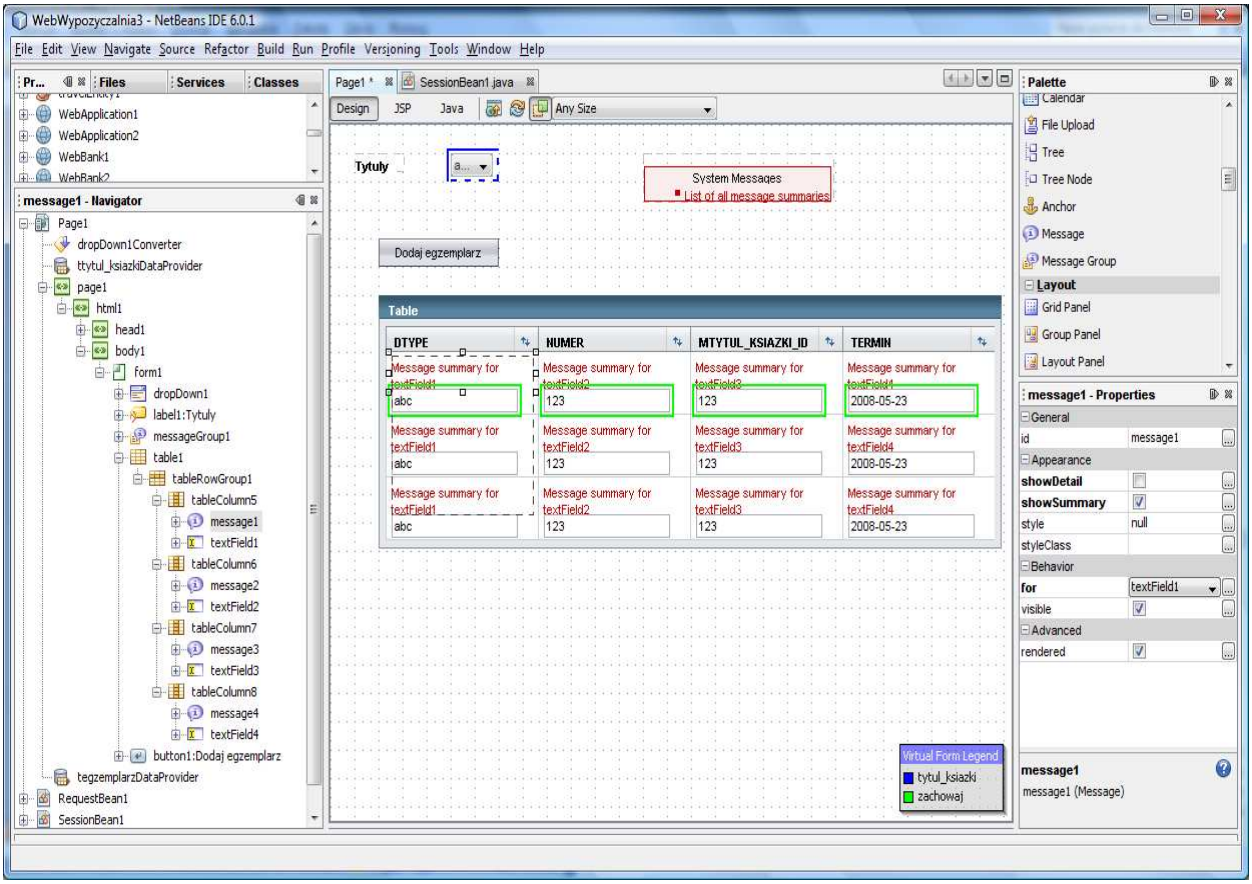
```
public void dropDown1_processValueChange(ValueChangeEvent event)
{
    try { //ustawienie wiersza w komponencie table1 danymi tabeli bazy danych skojarzonej z komponentem dropDown1 i
//wybranymi w tym komponencie
        getSessionBean1().getTegzemplarzRowSet().setObject( 1,
                                                dropDown1.getSelected());
        tegzemplarzDataProvider.refresh();
        form1.discardSubmittedValues("zachowaj");
        /*ta linia kodu powinna pojawić się po to, aby po wyborze nowego tytułu ukazały się nowe
        dane  w komponencie table1  */
        ttytul_ksiazkiDataProvider.refresh();
    //odświeżenie zawartości komponentu dropDown1
        } catch (Exception e) {
        error("Cannot switch to tytul ksiazki " +
                ttytul_ksiazkiDataProvider.getValue("TTYTUL_KSIAZKI.ID"));
        log("Cannot switch to tytul ksiazki " +
              ttytul_ksiazkiDataProvider.getValue("TTYTUL_KSIAZKI.ID "),e);
        }
}
```

9.6. Należy uruchomić aplikację za pomocą Build, Deploy i Run. Po wyborze klienta z listy dropDown1 w wierszu komponentu table1 ukażą się pozostałe dane tego klienta.



10. Należy przeciągnąć komponenty Message na każdy z komponentów typu TextField i za pomocą Ctrl+Shift przeciągnąć komponenty Message na TextField





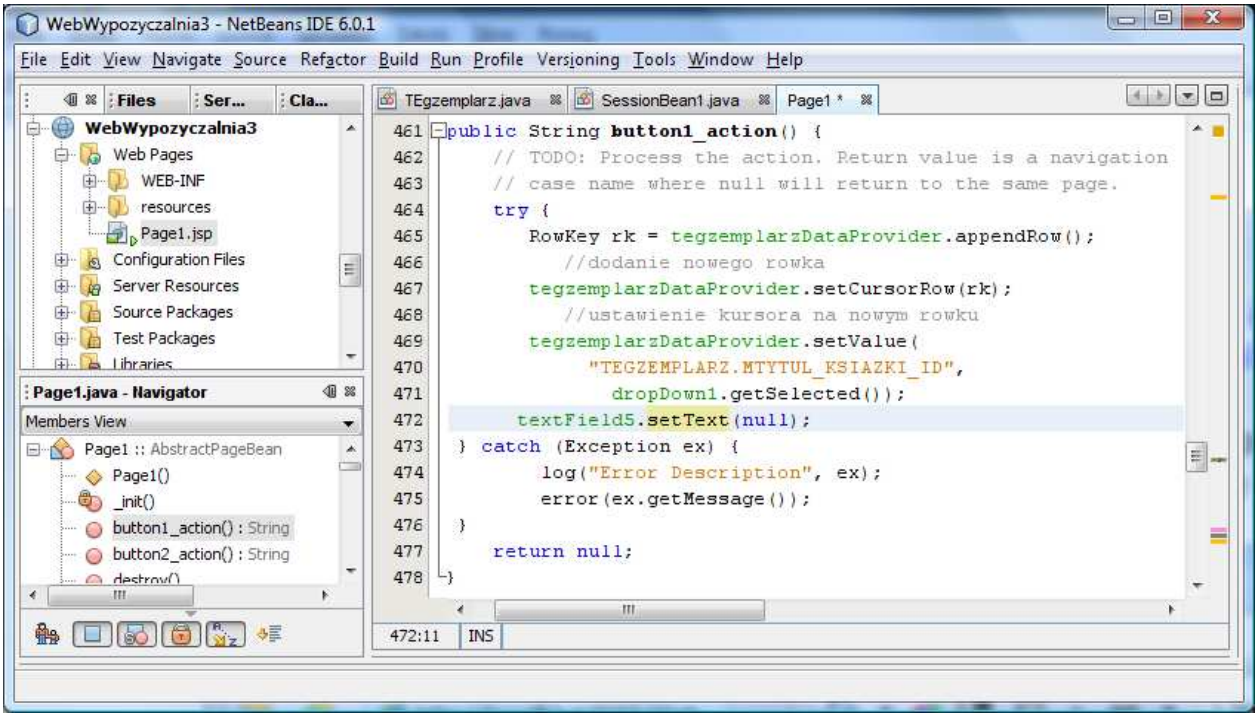
11. Wstawienie obsługi zdarzenia przycisku Dodaj egzemplarz

- Obsługa zdarzenia klawisza button1 Dodaj zakup. Po naciśnięciu klawisza button1 pojawia się nowy rowek w komponencie table1.

Plik Page1.java

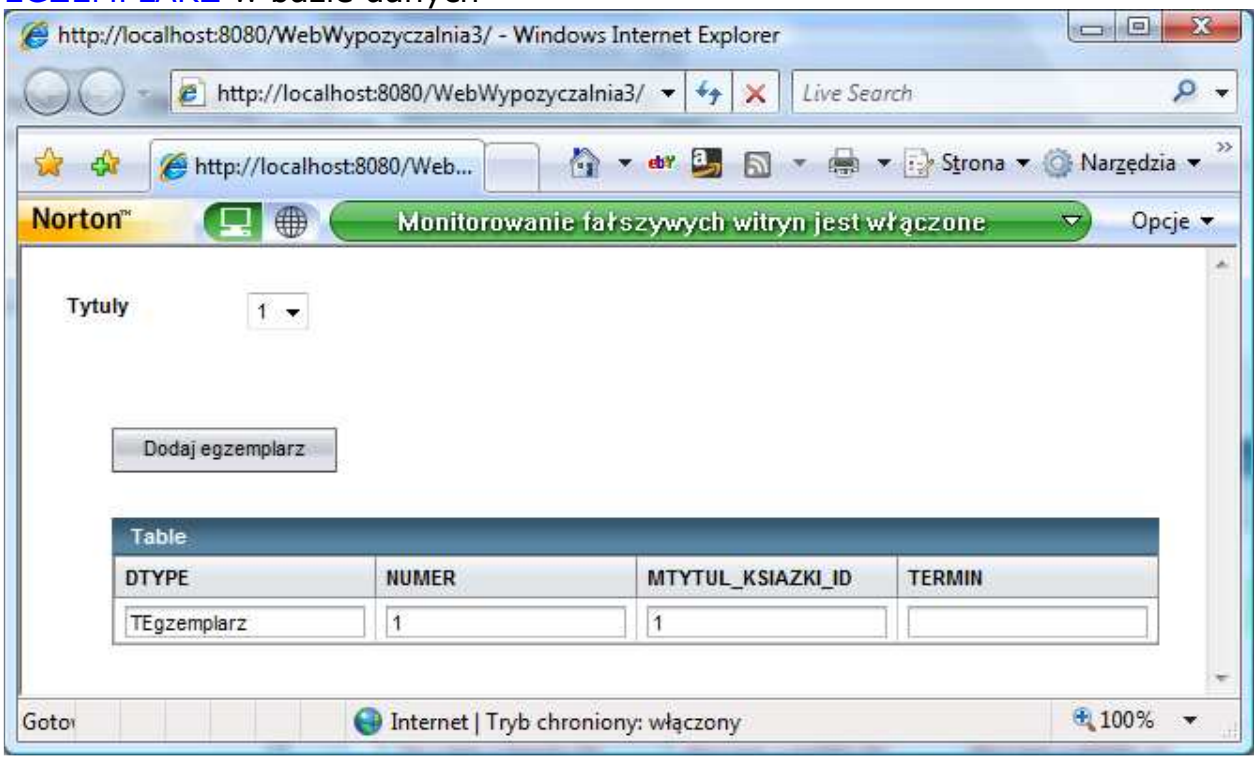
```
public String button1_action() {
        // TODO: Process the action. Return value is a navigation
      // case name where null will return to the same page.
  try {         //dodanie nowego rowka
        RowKey rk = tegzemplarzDataProvider.appendRow();
                //ustawienie kursora na nowym rowku
        tegzemplarzDataProvider.setCursorRow(rk);
                //powiązanie z tytułem książki wybranym z listy
        tegzemplarzDataProvider.setValue(
          "TEGZEMPLARZ.MTYTUL_KSIAZKI_ID", dropDown1.getSelected());
                //ustawienie pola tekstowego do wstawiania terminu
        textField5.setText(null);
     } catch (Exception ex) {
          log("Error Description", ex);
          error(ex.getMessage());
     }
        return null;
}
```

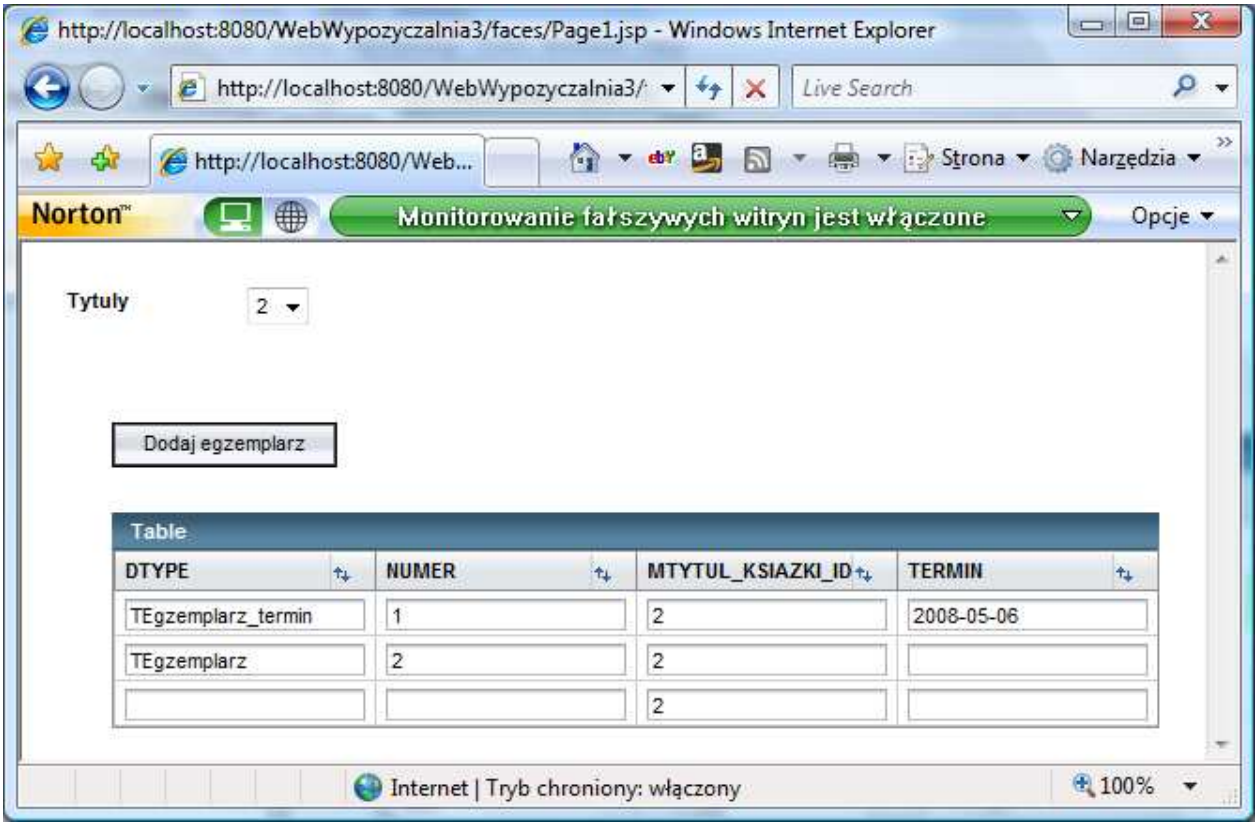
Należy wykonać import "import com.sun.data.provider.RowKey;" za pomocą klawiszy Ctrl+SHIFT+I



12. Należy uruchomić aplikację za pomocą Build, Deploy i Run. Po wyborze klienta z listy dropDown1 w wierszu komponentu table1 ukażą się jej zakupy.

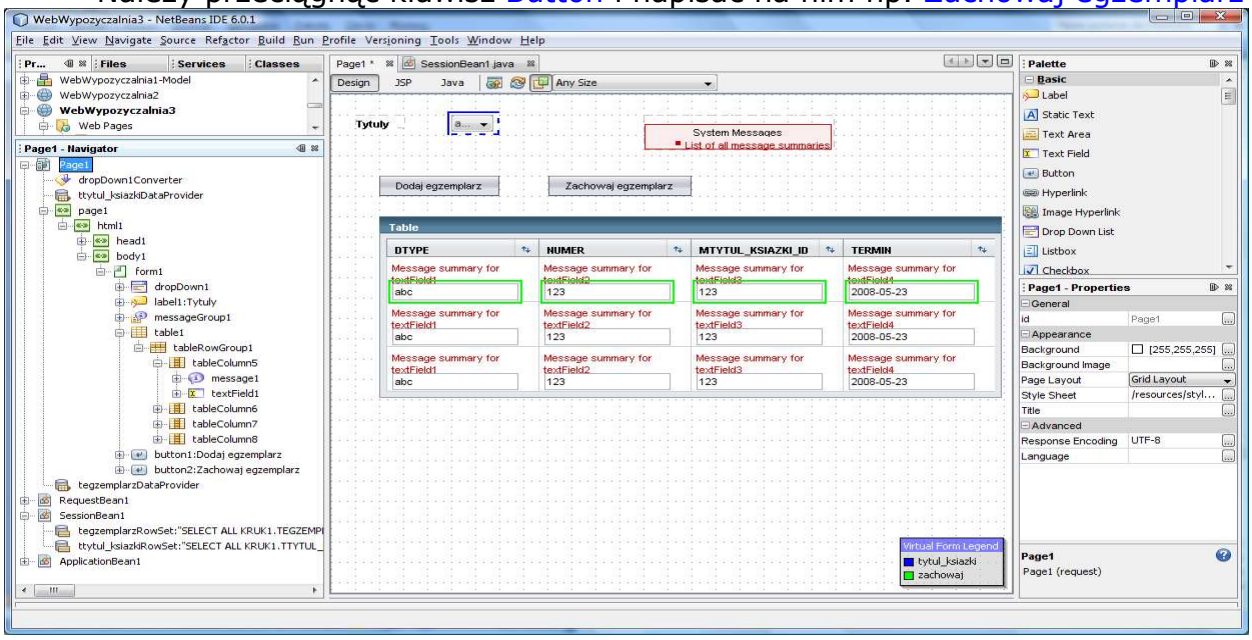
- Po kliknięciu klawisza Dodaj zakup ukaże się pusty rowek w komponencie table1. Dane wpisane tam nie są zapisywane do tabeli EGZEMPLARZ w bazie danych



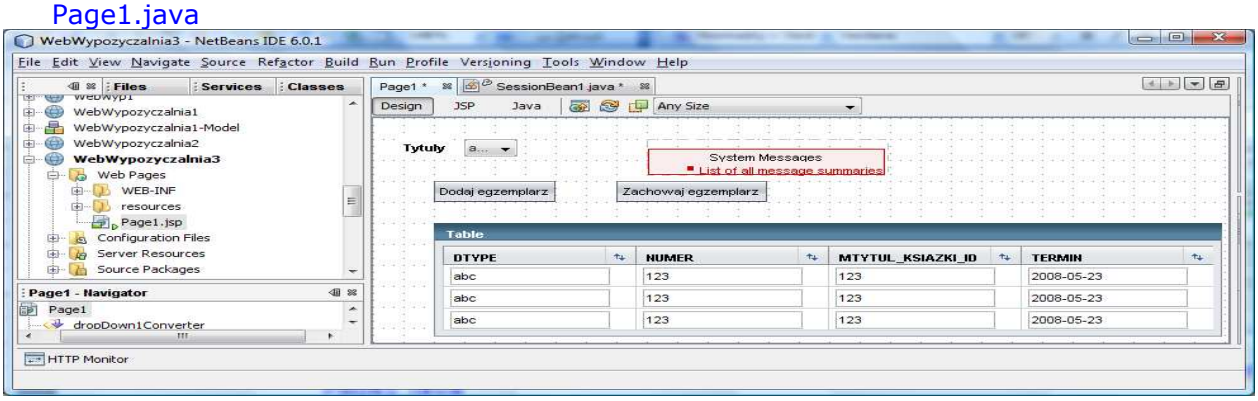


13. Zapisanie do tabeli nowego egzemplarza

- Należy przeciągnąć klawisz Button i napisać na nim np. Zachowaj egzemplarz

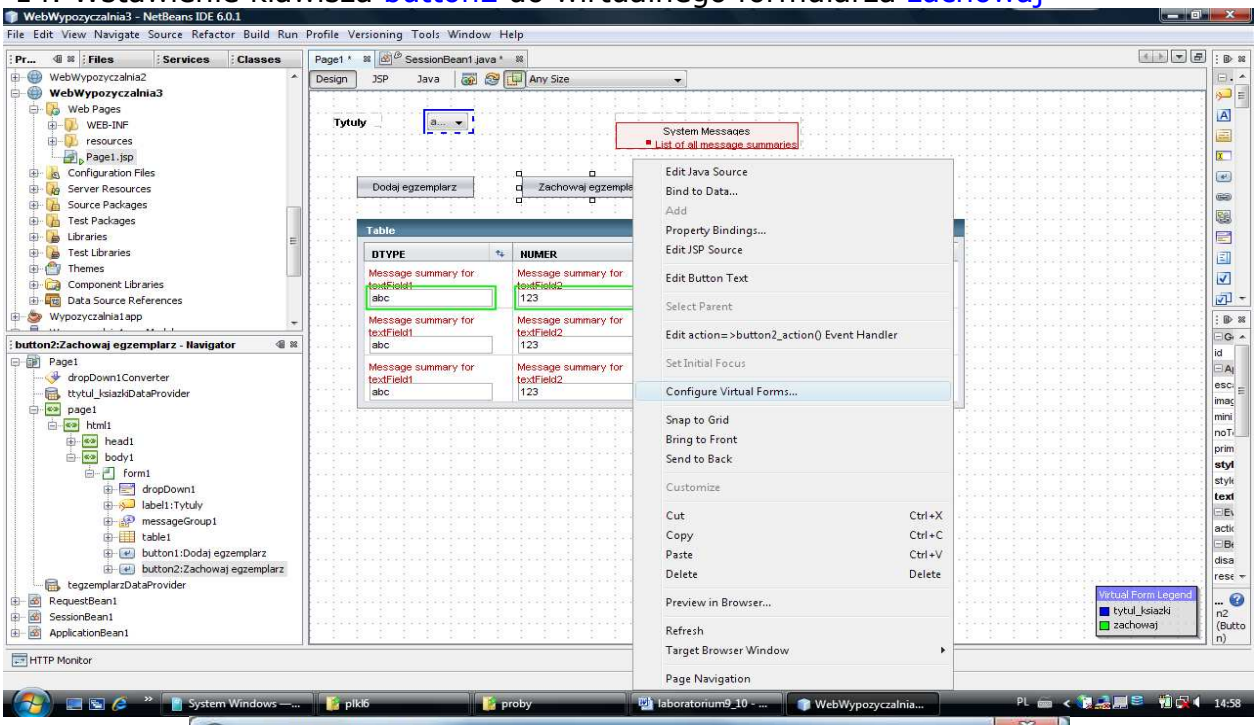


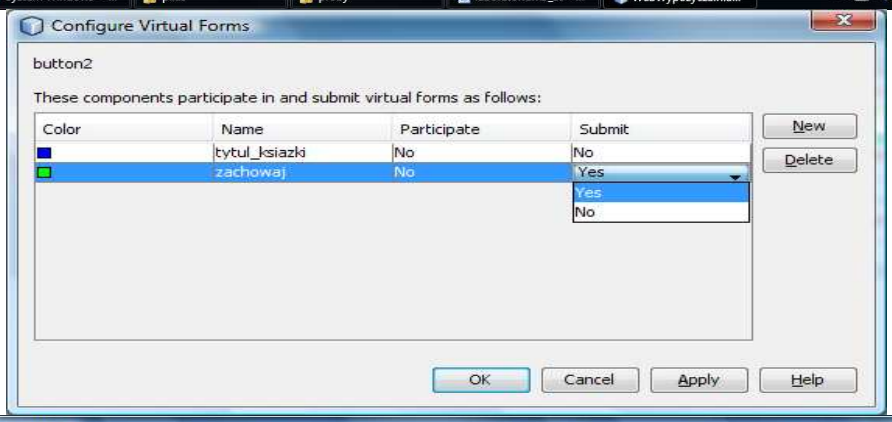
- Obsługa zdarzenia naciśnięcia klawisza button2 (Zachowaj egzemplarz) w pliku Page1.java

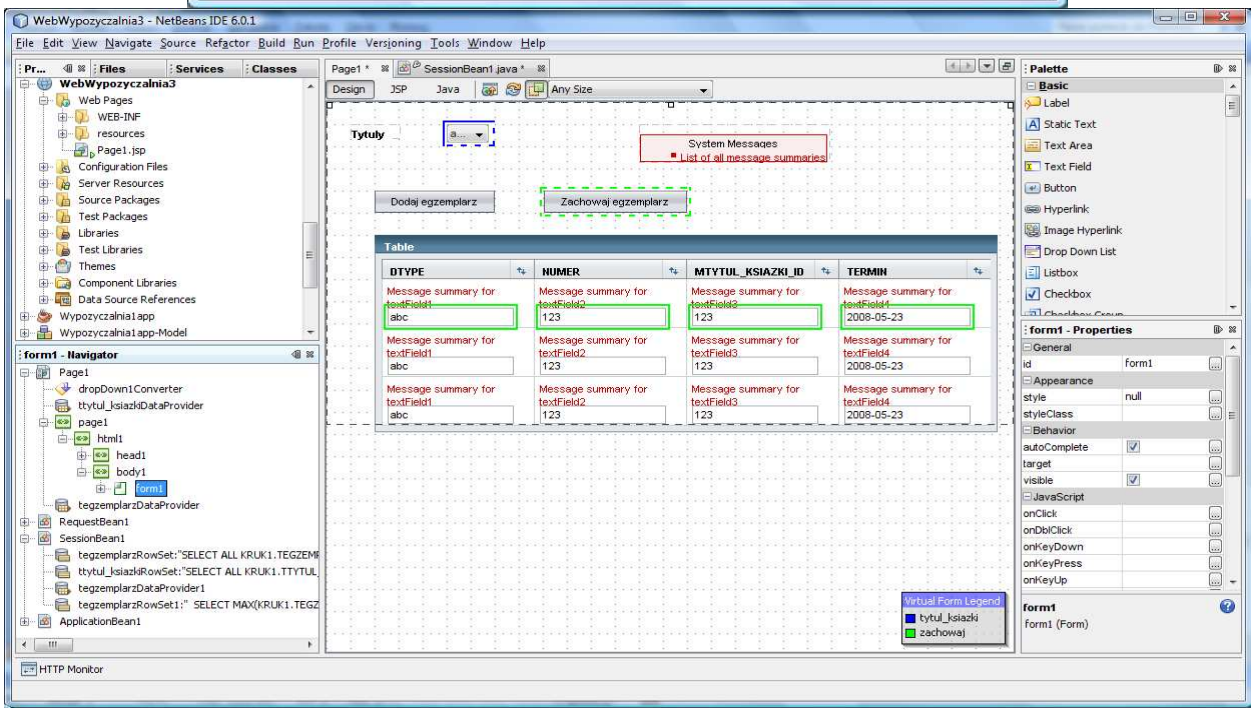



```
public String button2_action() {
    // TODO: Process the action. Return value is a navigation case name where null will return to the same page.
 try// Pobranie następnej wartości ID, korzystając z zapytania realizowanego przez tegzemplarzDataProvider()
  { if ( tegzemplarzDataProvider.getRowCount() > 0)
    { tegzemplarzDataProvider.cursorFirst();
      do { if  (((String)tegzemplarzDataProvider.getValue(
                                "TEGZEMPLARZ.DTYPE")).equals("TEgzemplarz"))
                tegzemplarzDataProvider.setValue("TEGZEMPLARZ.TERMIN",null);
           else if (((String)tegzemplarzDataProvider.getValue(
                              "TEGZEMPLARZ.DTYPE")).equals("TEgzemplarz_termin")&&
              tegzemplarzDataProvider.getValue("TEGZEMPLARZ.TERMIN")==null)
                 return null;//jeśli błąd w danych, zaniechanie utrwalania danych
         } while (tegzemplarzDataProvider.cursorNext());
   }//wprowadzenie do bazy wszystkich zmian, nie tylko wynikających z nowo wprowadzonego rowka
 tegzemplarzDataProvider.commitChanges(); //lub kilku nowych rowków
 ttytul_ksiazkiDataProvider.refresh(); //odświeżenie zawartości komponentu dropDown1
       } catch (Exception ex) {
          log("Error Description", ex);
          error("Error :"+ex.getMessage());
         } return null; }
```

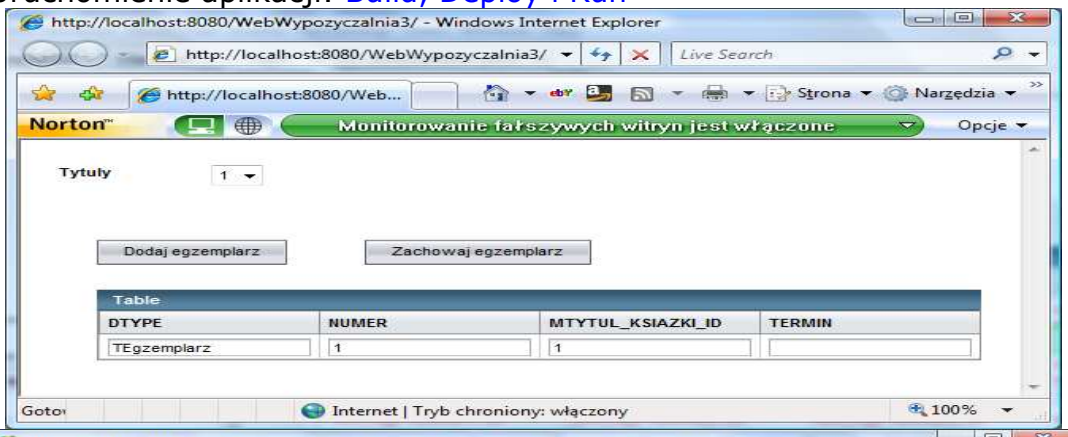
14. Wstawienie klawisza button2 do wirtualnego formularza zachowaj

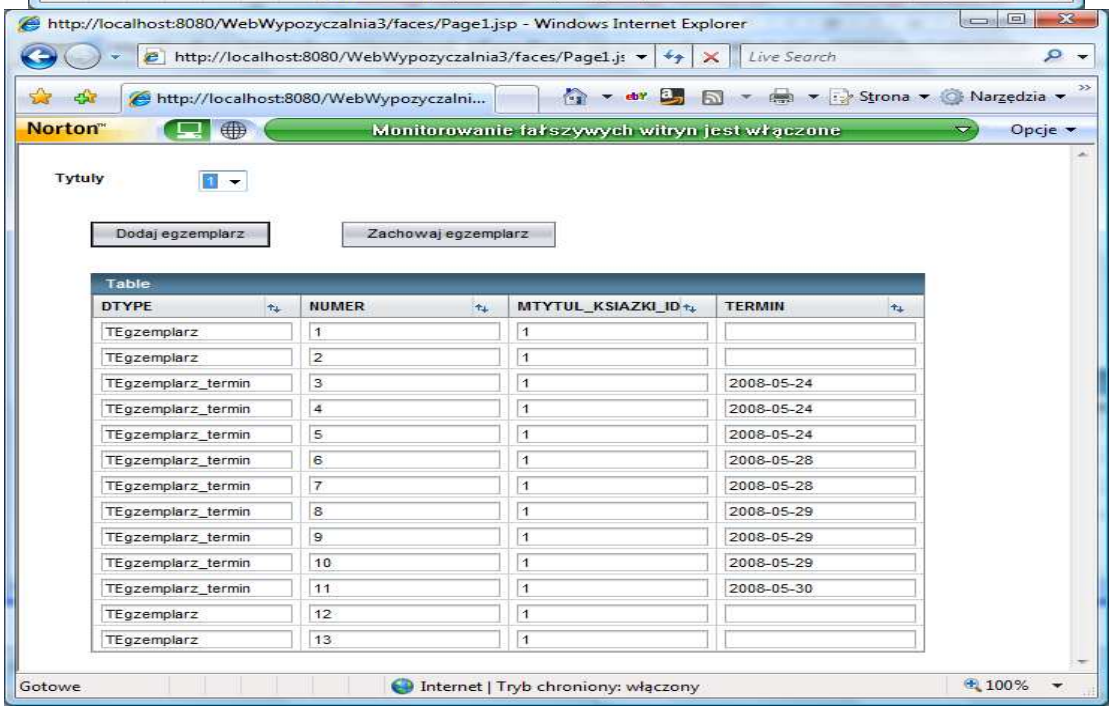






15. Uruchomienie aplikacji: Build, Deploy i Run





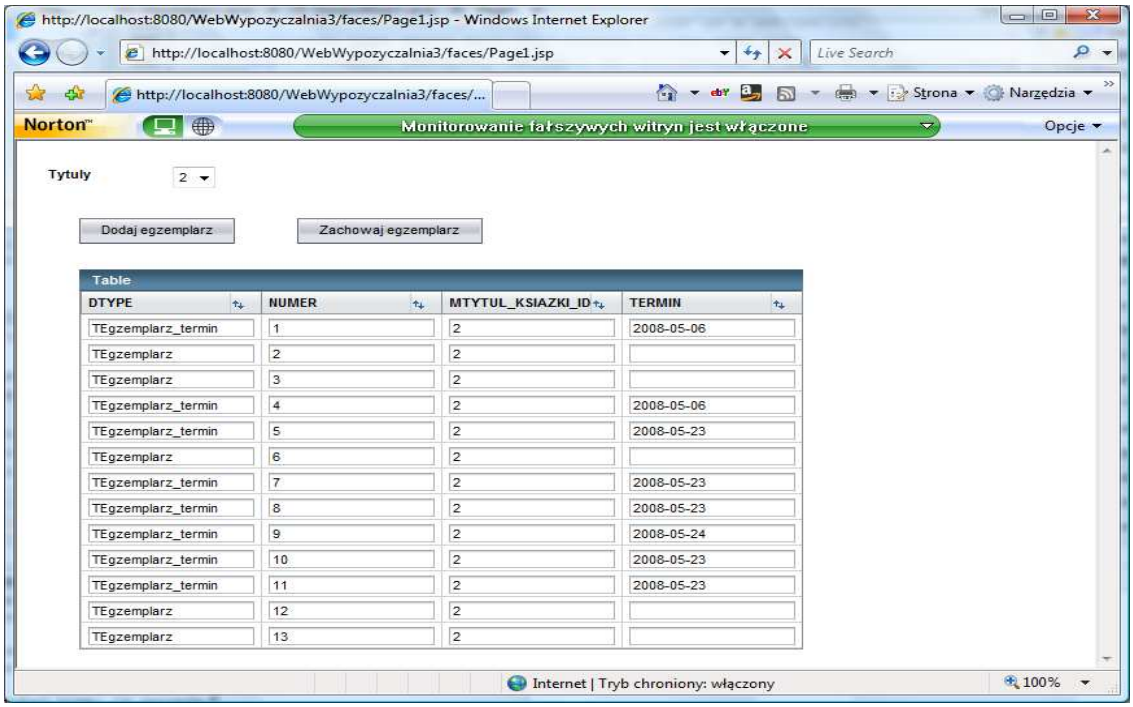


Table 1: Input Components

Component	Description	Palette Section
Text Field	An input field for a single line of text.	Basic
Text Area	An input field for multiple lines of text.	Basic
Password Field	An input field that echoes the input characters with a replacement character to mask the input.	Basic
Calendar	An input field with a pop-up calendar that allows the user to select a date.	Basic
Drop Down List	A drop-down menu, also referred to as a combo box.	Basic
Listbox	A list from which the user can select either one item or multiple items, depending on how the component is configured.	Basic
Checkbox	A single-character box that the user can either select (check) or clear.	Basic
Radio Button	A single radio button that the user can select (check).	Basic
Add Remove List	Two lists (one for available options, one for selected options) with buttons to move the options between the lists, and to order the selected options.	Composite
File Upload	A component with a text input field and a Browse button that displays a file chooser for specifying a file to upload. The application uploads the specified file when the user submits the page.	Basic

Table 2: Display Components

Component	Description	Palette Section
Static Text	Field for displaying text.	Basic
Label	Text field that can be associated with an input field and for which you can specify a weak, medium, or strong font style.	Basic
Image	Inline image.	Basic
Message	Text field that is linked to a specific component for displaying validation errors and other messages about that component.	Basic
Message Group	Text field for displaying runtime error messages, program generated error messages, and, optionally, validation errors and other messages about components that are on the page.	Basic
Alert	Displays an icon and informational text such as a warning, an error, or the successful completion of some event.	Composite
Page Alert	Similar to an Alert component, but is intended for displaying the icon and information on a separate page.	Layout

Table 3: Grouping Components

Component	Description	Palette Section
Checkbox Group	Displays two or more check boxes in a grid layout.	Basic
Radio Button Group	Displays two or more radio buttons in a grid layout and ensures that only one button can be selected at a time.	Basic
Table, Table Row Group, and Table Column	Displays data from a composite data type such as a database table or an array.	Basic
Tree and Tree Node	Renders an expandable list in a tree structure.	Basic
Tab Set and Tab	Displays different layouts on the same page. Also can be used as a navigational tool.	Layout
Grid Panel	Organizes the components within a layout of rows and columns.	Layout
Group Panel	Groups a set of components in flow layout mode.	Layout
Layout Panel	Use to group a set of components in flow layout mode or grid layout mode.	Layout
Property Sheet, Property Sheet Section, and Property	Lays out a single column of labeled components quickly, and divides the components into sections.	Layout
Breadcrumbs	Lays out a series of link components separated by right angle brackets (>).	Composite
Page Fragment Box	Groups components that you want to consistently display on two or more pages.	Layout

Table 4: Action Components

Component	Description	Palette Section
Button	Button that submits the associated form.	Basic
Hyperlink	Text field that submits a URL or submits a form.	Basic
Image Hyperlink	Image that submits a URL or submits a form.	Basic
Tree Node	Subcomponent of a Tree or Tree Node. A leaf tree node can optionally submit a URL or submit a form.	Basic
Tab	Subcomponent of a Tab Set or a Tab. A tab can optionally submit a URL or submit a form.	Composite